

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
30 November 2000 (30.11.2000)

PCT

(10) International Publication Number
WO 00/72623 A2

(51) International Patent Classification⁷: H04Q 11/00

W. [US/US]; 8 West Orleander Drive, Mt. Laurel, NJ 08054 (US). SCHAMING, Philip [US/US]; 41 Maple Village Court, Bernardsville, NJ 07924 (US). SULLIVAN, Raymond, Bradley [US/US]; 23 Camelia Court, Lawrenceville, NJ 08648 (US).

(21) International Application Number: PCT/US00/13774

(22) International Filing Date: 18 May 2000 (18.05.2000)

(25) Filing Language: English

(74) Agents: REYNOLDS, Leo, R. et al.; Hamilton, Brook, Smith & Reynolds, P.C., Two Militia Drive, Lexington, MA 02421 (US).

(26) Publication Language: English

(81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(71) Applicant (*for all designated States except US*): INC INTEGRATED NETWORK CORPORATION [US/US]; Bridgewater Corporate Park, 757 Route 202/206, Bridgewater, NJ 08807 (US).

(84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

(72) Inventors; and

Published:

(75) Inventors/Applicants (*for US only*): BLANK, Elijah [IL/US]; Bartle Court, No. 12C, Highland Park, NJ 08904 (US). ESKOLA, Paul, J. [US/US]; 76 Kim Lane, Long Valley, NJ 07853 (US). HALLERMEIER, W., David [US/US]; 2116 North Line Street, Lansdale, PA 19446 (US). HURLER, Joseph [US/US]; 61 Hobart Avenue, Summit, NJ 07901 (US). JACOBSON, Henry [US/US]; 16 Winterberry Circle, Piscataway, NJ 08854 (US). JOHNSEN, Gregory, F. [US/US]; 6 Delwood Road, Chester, NJ 07930 (US). MCGRATH, Bruce,

— *Without international search report and to be republished upon receipt of that report.*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

A2
3
WO 00/72623

(54) Title: LOOP ACCESS SYSTEM WITH GRAPHICAL USER INTERFACE

(57) Abstract: A local access system includes a variety of interchangeable resource modules under the control of a management unit over common bus. Each module communicates with the management unit over a message channel. The system provides a flexible interface for various network communication services, such as, narrowband POTS, BRI, DDS, Ethernet and SDSL services, as well as wideband DS1 and protocols, such as ATM, frame relay, IP. A management system is provided to enable a user to exercise, configure and test the system.

LOOP ACCESS SYSTEM WITH GRAPHICAL USER INTERFACE

RELATED APPLICATIONS

This application claims the benefit of U.S. Provisional Application No. 60/138,149, filed June 8, 1999; U.S. Provisional Application No. 60/137,984, filed June 5, 1999; U.S. Provisional Application No. 60/137,994, filed June 7, 1999; and U.S. Provisional Application No. 60/135,345, filed May 21, 1999, the entire teachings of which are incorporated herein by reference.

BACKGROUND OF THE INVENTION

A need exists for a flexible local access system for terminating a wide range of network communication services in central office, remote location and customer premises applications. The local access system provides an interface between network transmission facilities, switching systems and customer premises equipment. For example, both narrowband, e.g. POTS (Plain Old Telephone Service), BRI ISDN (Basic Rate Integrated Services Digital Networking), DDS (Digital Data Services), SDSL (Symmetric Digital Subscriber Line) services and wideband, e.g. T1, ATM and Frame Relay services need to be accessed, and this usually requires several separate and distinct dedicated access devices.

-2-

Moreover, due to the complexity of such an access system, a user interface is required that can enable an operator/user to exercise, configure and test the system.

SUMMARY OF THE INVENTION

The present invention provides a highly flexible local access system (LAS) and apparatus for interfacing a variety of network communication services. In a preferred embodiment the system comprises a plurality of resource modules which interface with commonly deployed network transmission facilities, switching systems and customer premises equipment. Each of the modules is monitored by and managed by a management unit called a shelf manger (SM) over a message channel and each resource module communicates with the SM via the message channel during start-up using a communication protocol over a one wire bus. Resource modules (RM's) can be cross-connected to any other resource module via the message channel to provide a full Time Division Multiplexed (TDM) time slot interchange (TSI) capability. A Graphical User Interface (GUI) Management System for the LAS is provided which aids the user in the performance of several functions such as configuration management, equipment, testing, status tracking and statistics reporting.

Each resource module is contained in a universal shelf assembly and each module may be removed and replaced while the LAS is energized, i.e. the modules are "hot swappable". A method and apparatus for providing clock and timing synchronization for the LAS is incorporated in the SM's which serves as the master of the system bus. The RM's have the ability to recover timing synchronization. Modules can be inserted and removed without the need to power down the system and without causing disruption or bit errors in active telecommunications circuits.

BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing and other objects, features and advantages of the invention will be apparent from the following more particular description of preferred embodiments of the invention, as illustrated in the accompanying drawings in which like reference characters refer to the same parts throughout the different views. The drawings are not necessarily to scale, emphasis instead being placed upon illustrating the principles of the invention.

-3-

Fig. 1 is a block diagram of a local access system (LAS) illustrating how it can be interconnected to various network resources.

Fig. 2 is a block diagram of a LAS configured for universal metallic access.

Fig. 3 is a block diagram of a LAS configured for point-to-point access.

5 Fig. 4 is a block diagram of a LAS configured in a STAR connection.

Fig. 5 is a block diagram of a LAS configured for an ADD/DROP connection.

Fig. 6 is a functional block diagram of a LAS in accordance with the invention.

Fig. 7 is a block diagram of a management unit (SM) in accordance with the invention.

10 Fig. 8 is a block diagram of a 4POTS resource module (RM).

Fig. 9 is a block diagram of a 2OCUDP resource module.

Fig. 10 is a block diagram of a 4BRI resource module.

Fig. 11 is a block diagram of a 2DSXI resource module.

Fig. 12 is a block diagram of an SDSL resource module.

15 Fig. 13 is a timing diagram of an INBAND data bus data frame format.

Fig. 14 is an illustration of a POTS DSO cross connect.

Fig. 15 illustrates the effect of Trunk Processing on the cross-connect.

Fig. 16 is an illustration of a signal channel assignment.

Fig. 17 depicts a GUI Management System (MS) Display.

20 Fig. 18 depicts a Zone A MS display.

Fig. 19 is a Zone A1 display which contains shelf descriptor and screen level information.

Fig. 20 is a Zone B MS display.

Fig. 21 is a Zone C MS display.

25 Fig. 22 is a view of the GUI display showing clock management of the SM configuration.

Fig. 23 is a view as in Fig. 22 showing selection of various sync options.

Fig. 24 is a view of the GUI display showing SDSL configuration.

Fig. 25 is a view as in FIG. 24 illustrating configuration of interface properties.

30 Fig. 26 is a view as in Fig. 25 showing configuration of the 4POTS module.

Fig. 27 is a GUI view showing notice to the user of a status change.

-4-

Fig. 28 is a GUI view illustrating how the display options are displayed.

Fig. 29 is a GUI shelf display view showing the cross-connections between modules.

Fig. 30 is a GUI equipment display user which shows the user all of the
5 connections associated with the interfaces of selected equipment.

Fig. 31 is a GUI interface display view.

Fig. 32 is a GUI display operational view.

Fig. 33 is a GUI Zone B SM faceplate and tabbed dialog box view.

Fig. 34 is a GUI Zone B RM faceplace and tabbed dialog box view.

10 Fig. 35 is a GUI shelf screen cross-connect view.

Fig. 36 is.a GUI display of showing the operation of the “connected to” buttons.

Fig. 37 is a view as in Fig. 36 of a cross-connect MAP interface screen.

Fig. 38 is a drill file down view of Fig. 37.

Fig. 39 is a GUI dialog box view.

15 Fig. 40 is a dialog box view as in Fig. 39 showing that a connection exists.

Fig. 41 is a GUI administrative view of the shelf screen display.

Fig. 42 is a GUI view as in Fig. 41 in which the resource module has been clicked.

Fig. 43 is a GUI interface administration view.

20 Fig. 44 is a diagram showing the context in which the MS software operates.

Fig. 45 is a top level data flow diagram of the data flow in the GUI.

Fig. 46 is a side view of an RM module showing access for a fiber connection.

Fig. 47 is a schematic block diagram of System Clock Generation and
25 Synchronization.

Fig. 48 is a timing diagram of Composite Clock Timing Relationships.

Fig. 49 is a schematic block diagram of Recovered Clock Source Provision Options.

Fig. 50 is a schematic block diagram of Locally Derived Clocks.

-5-

Fig. 51 is a timing diagram of Locally Generated Clocks Derived From The Backplane.

Fig. 52 is a timing diagram of Normal TDM Frame Cycle.

Fig. 53 is a timing diagram of Clock Handoff Frame Cycle.

5 Fig. 54 is a Clock Fallback and Reversion State Diagram.

Fig. 55 is a schematic block diagram of bus isolation circuitry.

Fig. 56 is a schematic block diagram of an in-rush current circuit.

Fig. 57 is a timing diagram which shows an initialization request.

Fig. 58 is a schematic diagram of backplane guide pins.

10 Fig. 59 is a schematic diagram of a guide pin receptacle.

Fig. 60 is a block diagram of a LAS configured for an SDSL connection to an integrated access device (IAD).

Fig. 61 is a block diagram of a LAS configured in a backplane extension application.

15 Fig. 62A shows a frame format with timeslots for communicating voice and data between an 2SDSL resource module and an IAD. Fig. 62B shows an 2SDSL frame format with timeslots for communicating data only. Fig. 62C shows an 2SDSL frame format for communicating clear channel data.

Fig. 63 is a block diagram of a LAS configuration illustrating trunk conditioning
20 in an 2SDSL application.

Fig. 64 is a block diagram of a two-line SDSL resource module.

Fig. 65 is a block diagram showing data, signaling, clock and synchronization signal connections in a portion of the two-line SDSL resource module of Fig. 64.

25 Fig. 66 is a block diagram of an upstream signaling portion of an FPGA in Fig. 64.

Fig. 67 is a block diagram of a signaling swizzler portion of an FPGA in Fig. 64.

Fig. 68 is a block diagram of an embodiment of a shelf-manager unit (SMU2).

Fig. 69 is a block diagram of a LAS configured with an Ethernet protocol resource module.

30 Fig. 70 is a block diagram of WAN encapsulation with respect to an Ethernet frame.

-6-

Fig. 71 is a block diagram of an embodiment of an Ethernet protocol resource module.

DETAILED DESCRIPTION OF THE INVENTION

The following acronyms or mnemonics are used throughout the text and are defined herein for reference:

	ATM Asynchronous Transfer Mode
	BRI Basic Rate Interface
	COT Central Office Terminal
	DDS Digital Data System or Digital Data Services
10	DLC Digital Loop Carrier
	EOC Embedded Operations Channel
	ESF Extended Superframe
	FPGA Field Programmable Gate Array
	FRAD Frame Relay Access Device
15	GUI Graphical User Interface
	IP Internet Protocol
	ISDN Integrated Service Digital Network
	LAS Local Access System
	LUNT Line Unit Network Termination
20	LULT Line Unit Line Termination
	NM Network Messages
	NTU Network Termination Unit
	PGTC Pair Gain Test Controller
	POTS Plain Old Telephone Service
25	RM Resource Module
	RT Remote Terminal
	SDSL Symmetric Digital Subscriber Line
	SLIC Subscriber Loop Interface Circuit
	SM Shelf Management Unit
30	TDM Time Division Multiplexing

TSI Time Slot Interchange

SYSTEM OVERVIEW

The local access system (LAS) 14 is a highly flexible, carrier class system capable of terminating a range of network services in both central office, remote 5 location and customer premises applications. The system is based on a modular shelf assembly 12 which can be equipped with a variety of interchangeable interface plug-in units called resource modules 10. See Figure 1. These resource modules provide interfaces to commonly deployed network transmission facilities, switching systems and customer premises equipment. By selection of the appropriate resource modules, the 10 system supports a wide variety of system configurations and provides features associated, inter alia, with conventional channel banks, data access multiplexers, Ethernet LAN switch, IPDSLAM and digital loop carrier systems.

The system:

- supports narrowband (e.g., POTS 19, BRI 17, DDS 15, SDSL 13) and 15 wideband 17 (e.g., DS1) services through a family of plug-in resource modules 10.
- provides high capacity in a small footprint supporting more than 90 POTS, 40 directly connected SDSL links, 80 ISDN BRI or 160 Ethernet circuits in a single 3U high shelf assembly.
- architecture accommodates resource modules (transport, protocol and 20 service).
- provides full time slot interchange at the DS0 level for efficiently managing circuit moves, changes and rearrangements. Cross-connects can be made between any two termination points, i.e., between one or 25 more DS0's via a back plane fabric.
- enables software control of administration, provisioning and maintenance with local and remote access.
- is fully compliant with applicable standards for deployment in central office or co-location applications.

-8-

- incorporates resource module slots which support narrowband or wideband resource modules 10 in line, or drop, configurations.
- is deployable in a variety of network configurations for applications in central offices, remote locations or customer premises.
- 5 • features resource modules 10 which are “hot swappable” and contain individual power converters for increased reliability
- supports timing options for internal, external composite clock and loop timing.

The shelf 12 is universal, in that it can be equipped with various resource
10 modules 10 and each module can be provisioned via management software downloaded from a management unit 21 (also provided on the shelf) to support a variety of configurations and functions, transport (trunks), service (customer interfaces) and protocol (ATM, frame relay). Because of this flexibility, service providers can bundle and deliver services tailored to meet their customer's requirements while efficiently and
15 cost effectively utilizing their own facilities and switching resources. For example, the system can readily be configured based on variables such as the services required by each individual customer (e.g., data only, POTS and data, fractional T1), by the premises physical arrangement (e.g., campus, multi-floor building), or by the carrier's access options (e.g., fiber, wireless, telco leased lines).

20 As will be illustrated in the following section, the system can perform central office/HUB functions such as DS0/DS1 cross connect and grooming, Digital Loop Carrier (DLC) remote terminal functions such as integrated switch access, and customer premises equipment functions such as add/drop multiplexing.

This section provides a brief description of four example configurations:
25 • Universal Metallic Access Configuration
• Point-To-Point Configuration
• Star Configuration
• Add/Drop Configuration

-9-

Universal Metallic Access Configuration

- As Figures 1 and 2 illustrate, the local access system 14 preferably consists of a single shelf 12 located in a carrier's hub or central office providing services to customer premises 18 via metallic loops 20 through conventional main distributing frame (MDF) 5 20A. In this configuration the system functions as a channel bank, providing metallic access 20 for the carrier's DS1 facilities 16A, a DLC central office terminal, providing cross-connect to the local exchange switch 16D, data equipment 16B, interoffice facilities 16C, and a DSLAM for the service provider's LAN.

Point-To-Point Configuration

- 10 As Figure 3 illustrates, in this point-to-point configuration the local access system (LAS) 14R at the remote location provides services via metallic interfaces 22. The remote LAS could be located in a cabinet or other outside plant enclosure or could be installed in an equipment room in a building. The LAS 14L at the central site provides the interface to the local exchange switch or other equipment as shown. When 15 voice or data switching equipment is not present at the central site, services can be groomed using the cross-connect functionality of the LAS 14 and then the traffic is transported over interoffice facilities to other locations. As shown in Figure 3, the interface between the LAS 14L and the central office equipment 16 can be DS1 or XDSL metallic depending on the type of interfaces available on the terminating 20 equipment 16. For example, voice traffic may be delivered to the switch via an integrated T1 interface and an ISDN circuit may be routed to the switch as a 2-wire Local Unit Network Termination (LUNT) U-interface. Likewise, private line and data traffic can be routed via DS1 carrier facilities Ethernet, or demultiplexed to baseband for local termination. It is important to note that the DS1 interfaces 25 on the LAS are 25 independent of transport type and can be carried over fiber, wireless or metallic facilities 26.

-10-

Star Configuration

As Figure 4 illustrates, the LAS 14 can be used in a Star Configuration with services being delivered to three remote locations A, B, and C via local access facilities 26. The LAS 14C at the central site provides groomed DS1 interfaces 25A for efficient 5 utilization of switches, multiplexers and interoffice facilities, and provides capabilities for easy moves, changes, and rearrangements. In this configuration DS1 interfaces 25 are shown on the local access side of the unit, however, LAS 14C could also provide metallic baseband interfaces with use of an appropriate resource module.

Add/Drop Configuration

10 As Figure 5 illustrates, the LAS 14 can be applied in an Add/Drop application. In this figure Locations A, B, and C could represent different floors in a high rise office building or different buildings in a campus environment. In such a configuration, circuits from several nearby locations can be aggregated and groomed to permit efficient utilization of local access transport facilities.

15 Physically, as shown schematically in Fig. 6, an LAS 14 consists of one or more shelf assemblies 12, each equipped with a common management or control unit called a Shelf Management Unit (SM) 21. The shelf provides slots for a plurality of plug-in resource modules 10 as well as one SM 21.

LAS Shelf Assembly

20 The shelf 12 (Fig. 6) is a compact unit sized for standard rack mounting that contains 26 slots, 25 of which may contain resource modules and one for containing an SM 21. The shelf includes a high speed distributed backplane (not shown) capable of supporting up to 1024 DS0 cross-connects. The distributed nature of the shelf allows connectivity between any two circuits in the system. Also included in the shelf is a 25 control or message channel 30 which enables out-of-band communications from the SM to the individual modules in the system over a one-wire interface during start-up. This channel also serves such purposes as provisioning, alarm reporting, and performance monitoring.

-11-

Connections on the backplane allow for -48 Vdc, -48 Vdc return, frame ground, composite clock and dry alarm contacts.

Shelf Management Unit (SM) 21

The SM 21 is the central controller for all system shelf modules. The SM 5 provides the common logic functions for all shelf modules including system timing, alarm reporting, software configuration and performance monitoring.

The SM monitors each resource module 10 via a message channel 30 and provides the interface for software downloads and remote and local operator access.

Resource Modules 10

10 Resource modules can be plugged into any universal slot in the shelf 12 to provide interfaces for a variety of services.

The system supports a broad range of customer services while providing the flexibility to the service provider to support a large variety of configurations in a cost effective manner. These configurations can support diverse applications which utilize 15 widely deployed transport media such as fiber optic cables, metallic pairs and wireless systems.

There are three classes of resource modules: service, transport and transport protocol resource modules. Service related modules include POTS, DDS, and ISDN cards for service offerings. Transport related modules include T1, xDSL, OC3, etc. for 20 transport connections. Resource modules related to the transport protocols include GR303, IP, TDM, ATM, FR, etc. Any resource module can be inserted into any one of 25 card slots for their operation. Through SM control, the LAS shelf 12 supports the digital cross connect and grooming functions among different resource modules for further optimization of transport bandwidth. Multiple shelf operation is supported 25 through inter-transport resource module connections. The architecture allows smooth and painless migration of multiple service offerings on today's circuit switched network to tomorrow's IP network without physical alterations. This is accomplished through software provisioning via the message channel and by the future addition of an IP

-12-

resource module in the shelf. Thus, the LAS is horizontally and vertically scalable, from two circuits to thousands of circuits.

The total throughput capacity of a single shelf is 130 Mbps, which exceeds DS3 capability.

- 5 As shown in Figure 6, a resource module 10 in any slot has full access to the full 2048 DSO bandwidth available on the backplane to assure full cross connect capability. Also, all units are monitored by and communicate with the SM via the control channel 30. For start-up and shut down operation a one wire bus 13 allows communication between the SM and each resource module. Each type of resource module 10 provides 10 the external interfaces 32 for cross-connection to external equipment via fully connectorized cables (not shown) which are provided with the shelf 12.

The LAS architecture provides a number of unique features, namely:

- Channel(s) associated with any resource module 10 can be cross-connected to any other module and each channel unit in a module can 15 read or write from any time slot on the backplane TSI/Data Bus 42, providing full time slot interchange (TSI) capability i.e.
- Only one common equipment module 21 is required per shelf, eliminating the need for a separate common control shelf and reducing the overall system cost.
- A resource module 10 can be inserted in any slot, providing the 20 flexibility for the same shelf to be utilized in any configuration required to support a customer application.
- All modules are formed on printed circuit board cards which are the same physical size from a four circuit POTS card to an OC-3C 25 transceiver.

Maintenance and Alarms

The SM provides centralized common maintenance and alarm functions for the system. These functions include:

- Maintaining, in non-volatile memory, common equipment settings and 30 resource module configurations.

-13-

- Storing and executing maintenance commands such as loopbacks initiated via the operator interface.
- Detection of module insertion or removal.
- Detection of a change in the external subscriber interface.
- 5 • Activation of metallic audible and visual alarm contacts which may be used for connection to external alarm systems.

This completes the system overview section of the specification. Next, the various units or modules will be described in more detail along with the signaling and communication buses interconnecting or providing communication paths between
10 modules.

Shelf Management (SM) Unit 21 (Fig. 7)

The SM is the system's intelligent controller unit. The SM performs the centralized tasks of initialization, fault monitoring, system timing and synchronization, alarm reporting, maintenance operations, operator interfacing, and software
15 downloading for all LAS resource modules 10. Configuration and optioning of the SM 21 and resource modules 10 is accomplished through use of the operator interface 53 located on the SM's faceplate.

The SM's microprocessor 40 provides static RAM, non-volatile RAM and flash memory for program storage and execution. The non-volatile RAM stores the system's
20 operational database containing unit configuration data. Each module 12 is provided with status indicators operable by the respective microprocessors in each module.

The Message Channel 44 is the link that communicates over the bus 42 with the resource modules, transporting configuration data, performance monitoring data, status condition data, alarm information, maintenance commands, and software program
25 downloading. The Message Channel also coordinates switchover of reference clock sources if the primary source fails.

The Clock Recovery and System Clock circuit 46 provides centralized control and distribution of the timing synchronization clocks. This circuit derives system timing from an external central office Composite Clock received on line 48 for Office
30 Timed situations via Backplane I/O 45. The composite clock is a 64 kbps, all ones

-14-

bipolar return-to-zero signal with a bipolar violation every eight pulses. (See Bellcore GR-378 (Ref. 6) and ANSI T1.101 (Ref. 1) incorporated herein by reference.) Timing can also be derived from a designated DS1, BR1, SDSL circuit (i.e., loop timing over line 47). In either case, the circuit 46 provides a mechanism for smooth switchover in
5 the event of a primary timing source failure. Also, the circuit 46 has an internal oscillator that can be engaged to provide synchronization via line 41 should the primary source fail.

The Alarm Processing Circuit 50 provides the relays and supporting logic for alarm contacts that connect to external alarm panels via bus 43 and Backplane 45. Both
10 visual and audible contact relay sets are provided for Minor (loss of one or more reference sources excluding a minimum level), Major (minor alarm persists for 24 hours or loss of all reference sources) and Critical alarms (severe condition exists and immediate response is requested).

A DB9 connector port 52 for the operator interface is located on the SM
15 faceplate. This port is an asynchronous serial link (i.e., 19200,8,1,N) that is accessible using a VT100-type craft access terminal or a PC with the appropriate terminal emulation software. This interface, which allows the operator/user to manage system resources, enables connection to:

- A menu-based control screen from which the operator can monitor, configure, and perform maintenance on all system components; and
- A reporting system that detects autonomous system events. These events consist of alarms, as well as conditions of common equipment, resource (i.e., data-bearing) module equipment, and termination points.

Resource Module Types

25 Several different types of resource modules can be provided in the LAS including, inter alia, the following:

-15-

1. A Quad POTS Module (4POTS) 60 shown in Fig. 8 is equipped with four independent POTS channels 61.

5 This module is used for long loop applications with bulk ringing available from an external ringing generator (not shown). In addition to conventional voice or modem applications, each 4POTS module 60 can service foreign exchange (FX) or other applications for which the system is used in either pair gain or digital loop carrier applications.

10 The 4POTS module 60 provides analog voice service using well known loop start signaling. The 4POTS module also supports On-Hook Transmission (OHX) for applications such as meter reading and call party identification. Other supported features include distinctive ringing and forward call disconnect.

The 4POTS module 60 includes a microprocessor 62 with static RAM and FLASH memory for program storage and execution.

15 Message Channel 64 is the command and control link used for module to module communication. The Message Channel transports configuration data, performance monitoring data, status/conditions, and maintenance commands to and from the back plane I/O 66.

20 The individual channel circuitry 61 performs functions that pertain to each of the four POTS channels 1-4. Transient protection circuits 71 protect each 4POTS channel 61 from lightning and power cross high-voltage transients on the subscriber line. The CODEC 76 performs the analog-to-digital and digital-to-analog conversion from the Backplane I/O 66 to the Data Bus 63 and the POTS interface 68. The SLIC (Subscriber Loop Interface Circuit) 74, see References 25/26, performs current feed and all supervision for the subscriber line.

25 The 4POTS module 60 receives provisioning information (e.g., on hook transmission, etc.) from an external console workstation via the system operator interface (not shown). 4POTS provisioning information is stored within the system and downloaded to the microprocessor 62 whenever a new 4POTS is inserted or a download command is entered. Data rate conversion is performed

-16-

in unit 75 to match the interface data rate of the 4POTS module to that of the internal RM data bus 63.

The 4POTS module 60 and the 2DSXI module 110 (subsequently described in connection with Fig. 11) each include a signal extraction and insertion unit 65 and 123 respectively, which provides a universal means of sending signaling information between termination points using the backplane data buses 63 and 128 respectively in conjunction with information received from the SM 21. Since the two units 65 and 123 are substantially identical, only unit 65 will be described in detail herein. Unit 65 includes a FPGA (not shown) which provides a signaling multiplexer function and logic function consisting of gathering signaling information from multiple termination points and formatting the information as depicted in Table 9 supra.

POTS Expansion

Variants of the 4POTS module are included in the LAS family to provide expanded voice services as described below:

The 4POTSR is used to provide voice services in short loop applications and contains an on-board ringing voltage generator and its design is strongly based upon the 4POTS.

The 4POTST, identical in hardware architecture as the 4POTS, has functions to support ground start supervision signaling and it also enables the LAS to operate with voice switches that work with the TR08 protocol.

2. A Dual OCU Data Port Module (2OCUDP) 90 shown in Fig. 9.

The 2OCUDP 90 is equipped with two independent, 4-wire DDS interfaces 92 with the Backplane I/O 77. The 2OCUDP 90 is typically used to support network access to standard DDS and Switched 56 services via Backplane I/O 77.

Each 2OCUDP interface 92 can be configured for either standard DDS operation at data rates of up to 64 kbps (up to 56 kbps with secondary channel service) or Switched 56 kpbs operation.

-17-

The 2 OCUDP 90 includes a microprocessor 94 with static RAM and FLASH memory for program storage and execution. A signal processing block 91 provides loop coding violation processing and storage, and performance monitoring functions.

5 The Message Channel 96 is the command and control data link used for module to module communication. The Message Channel transports configuration data, performance monitoring data, status/conditions, and maintenance commands to and from Backplane I/O 77.

10 The individual channel circuitry 99 performs functions that pertain to each of the two OCU data channels 1 and 2. Sealing current 97 of 47 millamps is applied. Transient protection 95 protects the 2OCUDP from lightning and power cross high-voltage transients on the subscriber line. The DDS Transceiver unit 89 performs all line equalization and data extraction in the receive direction and provides pulse shaping and filtering in the transmit direction. The Signal Processing circuits 91 provide functions for loop quality monitoring, loopback detection and generation, error correction, zero code suppression, signal recovery, and call progress monitoring. Data rate conversion is performed in unit 87 to match the 2OCUDP interface data rate to that of the data bus 93.

15 As with all the resource modules, the 2OCUDP 90 receives provisioning information (e.g., data rate selection, sealing current, etc.) from an external console workstation (not shown) via the system operator interface. 2OCUDP provisioning information is stored within the system and downloaded preferably from the SM into the microprocessor 94 whenever a new 2OCUDP is inserted or a download command is entered.

20 Each 2OCUDP channel independently reports performance monitoring and threshold-crossing conditions to the SM. Each 2OCUDP channel can execute both latching and non-latching loopback conditions. (These software-controlled diagnostic tools allow evaluation of data path integrity from both the central and subscriber site. For further information concerning loopbacks see U.S. 5,553,059 incorporated herein by reference.)

-18-

3. A Quad Basic Rate Interface Module (4BRI) 100 shown in Fig. 10.

The 4BRI 100 is equipped with four Integrated Services Digital Network (ISDN) Basic Rate Interfaces (BRI) 102 using 2B1Q modulation. Each 4BRI interface 102 can provide 2B+D service or any channel subset using 1, 2, or 3 DS0 channels from the carrier system, depending upon the ISDN service being transported. The 4BRI 100 is typically used to provide network access via ISDN services to end users for combined high-speed data and voice transmission needs over a single circuit. All provisioning and diagnostics are software controlled via the system's operator interface (not shown).

10 The 4BRI incorporates a microprocessor 104 with static RAM and FLASH memory for program storage and execution. The microprocessor 104 provides embedded operations channel (eoc) processing and storage, cyclic redundancy check (CRC) generation and verification, and performance monitoring functions.

15 The Message Channel 106 is the command and control data link used for module to module communication. The Message Channel transports configuration data, performance monitoring data, status/conditions, and maintenance commands to and from backplane I/O 109.

20 Each 4BRI channel 102 provides an ISDN 'U' interface 105 and functions as either a Line Unit Line Termination (LULT; i.e., interface to subscriber) or a Line Unit Network Termination (LUNT; i.e., interface to network switch). Each 4BRI channel, configured as a LULT, provides optional sealing current from unit 108.

25 A transient protection circuit 101 protects the 4BRI from lightning and power cross high-voltage transients on the subscriber line. The ISDN 'U' transceiver 103 performs all 2B1Q line signal encoding and decoding.

30 The 4BRI 100 receives provisioning information (e.g., LUNT/LULT, time slots, etc.) from an external console workstation via the system operator interface (not shown). 4BRI provisioning information is stored within the system and downloaded to microprocessor 104 whenever a new 4BRI is inserted or a download command is entered.

-19-

Each 4BRI channel 102 independently reports performance monitoring and threshold-crossing conditions to the SM. This allows performance data storage at both the 4BRI RM and system level. Rate conversion unit 107 is provided as with other modules.

5 4. A Dual DS1 cross connect RM (2DSX1) 110 is shown in Fig. 11.

10 The 2DSX1 RM 110 is equipped with two channels 112 to interface to DS-1 short-haul line facilities. Each interface operates at a range of up to 655 feet from the DSX cross-connect. The 2DSX1 is typically used to support network access for data circuits, voice circuits requiring D4 signaling, and drop and insert configurations for circuit grooming.

15 As with other modules, the 2DSX1 Module 110 incorporates a microprocessor 114 with static RAM and FLASH memory for program storage and execution. This microprocessor provides ESF facility data link processing and storage, cyclic redundancy check (CRC) generation and verification, and performance monitoring functions.

20 Message Channel 116 is the command and control data link used for module to module communication. The Message Channel transports configuration data, performance monitoring data, status condition data, and maintenance commands to and from the Backplane I/O interface 115.

25 A Clock Recovery circuit 118 maintains the clock synchronization of the 2DSX1 module. In central office timed systems, this circuitry receives clock inputs from the system via bus 117. In loop timing situations, the synchronization of the system is sourced from an operator-defined DS-1 interface. The recovered clock signal is routed to the system via bus 119 for conditioning and backplane distribution to all other DS-1 links.

30 The individual channel circuitry 112 performs functions that pertain to each of the two 2DSX1 lines. The line transceiver 120 provides the interface to the DS-1 pairs. Line coding (for example AMI and B8ZS) is introduced as configured to the data stream. Transient protection circuits 122 shields the 2DSX1 from lightning and cross high-voltage power transients on the subscriber

-20-

line. A framer 124 provides the frame formatting and signaling pulses for insertion into the data stream. Data rate conversion is performed by unit 126 to match the DS-1 data rate to that of the data bus 128.

5 Each 2DSX1 interface independently reports performance monitoring and threshold-crossing conditions to the system. This allows performance data storage at both the 2DSX1 and system level. Each 2DSX1 link can execute an operator initiated loopback. This allows evaluation of data path integrity from both the central and subscriber site. Network initiated extended superframe (ESF) line and payload loopbacks are also supported.

10 Alternate embodiments of the 2DSX1, called the 2T1 and 2T1t, provide all of the functionality of the above (DSx1) while adding the capability of supporting long haul (distance) connectivity with the addition of a CSU (Channel Service Unit) function. The 2T1 Rms provide signal levels that permit the LAS to interface directly with a CSU on the T1 side of repeaters.

15 Other variants of this RM provide different signaling types (eg., D4, TR-08) and formats (SF, ESF).

5. A Symmetrical Digital Subscriber Line module (SDSL) 130 shown in Fig. 12.

20 The SDSL module 130 is a high speed digital transport device, operating in full duplex mode over a single pair of twisted copper wire using 2B1Q modulation. The SDSL is typically used to provide network access to end users whose data transmission needs have outgrown 56/64 kbps services, but who do not require a full T1. The SDSL module communicates remotely with an SDSL or an SDSL/FRAD customer premises unit.

25 The SDSL Module utilizes a microprocessor 132 with static RAM and FLASH memory for program storage and execution. This microprocessor provides EOC message processing and storage, cyclic redundancy check (CRC) generation and verification, and performance monitoring functions.

A Message Channel 134 is the command and control data link used for module to module communication. The Message Channel transports

-21-

configuration data, performance monitoring data, status condition data, and maintenance commands to and from backplane I/O 140.

A system Clock Recovery circuit 136 recovers synchronization for data bus operation.

- 5 The channel circuitry 137 performs functions that pertain to the SDSL line interface. Sealing current 131 is available as a provisioning option. Transient protection circuit 133 shields the SDSL from lightning and cross high-voltage power transients on the subscriber line. The SDSL transceiver 135 performs all of the 2B1Q line signal encoding and decoding required. Data rate conversion is performed in unit 139 to match the SDSL data rate to that of the data bus 138.
- 10 15 The SDSL receives provisioning information from an external console workstation via the system operator interface. SDSL provisioning information is stored within the system and downloaded whenever a new SDSL is inserted or download command is entered.

Backplane Data Bus

- The LAS is provided with a Backplane Data Bus 42 (Fig. 6) which preferably consists of 16 synchronous time division multiplexed (TDM) serial data lines (SD0 to SD15), each providing 8.192 Mbps or 128 digital signal level 0 (DS0s) for a total LAS 20 bandwidth of 131 Mbps or 2048 DS0's. A full duplex channel consists of 2 DS0 time slots.

- 25 Each RM channel unit has the capability of writing to or reading from any timeslot on the backplane. This provides full time slot interchange (TSI) capability. The channel units have access to a total of 16.384 Mbps up to 256 DS0s. The channel units also have the ability to bundle multiple DS0s into hyperchannels.

- The data lines are synchronized as shown in Fig. 13. SCLK and FSYNC pulses are used for this purpose. Data is shifted onto the bus on the rising edge of SCLK pulse. Data is shifted off of the bus either on, or after, the falling edge of SCLK pulse. The time slot counter is reset during the FSYNC pulse and the time slot counter is 30 incremented or reset on the rising edge of SCLK pulse. A "frame" is defined as 125 μ s.

-22-

Bit 1 of the timeslot is transmitted first and is designated as the sign bit for PCM data, bit 1 of a DS0 byte.

The following describes the protocol and messages that are used to communicate between an external NM (Network Managers) (not shown), SMs 21 (Shelf Managers),
5 and the RM 10 (Resource Modules) over the message channel. These messages are used to transfer information such as: configuration, performance, status, events, alarms, and software. This information transfer is bi-directional. Any NM, SM or RM can initiate a message.

- All intrashelf communications are accomplished via the message channel.
- 10 Internodal communications will use other communications vehicles. Any module has the capacity to talk to any other module. Typically, communications will be between NMs and SMs, or SMs and RMs. RM to RM interworking is rare.

Messages can be initiated by any module in the shelf. The RMs would most likely create a message in order to report an event or request a service. SMs would
15 typically send messages to provide information or request that some action be performed. NMs would perform similar functions. NMs, SMs and RMs are not limited in any way to these specific scenarios.

- All intermodule communication is defined by the messages described below.
- Each message contains a header section and a data section. The header section contains
20 common fields for all message types and subtypes. The data portion contains information that is specific to the message type and subtype. All message types must be supported by all RMs, SMs and NMs. Message subtypes may or may not be supported depending upon the specific RM. All subtypes must be supported by all of the SMs and NMs.
- 25 All messages conform to a common template. Each message can consist of up to five hundred and twelve (512) bytes of information. A header and a data section are the two parts of this template.

A sixty four (64) byte header is at the beginning of each message. The fields in this header are common to all message types. Each field in the header must contain a
30 logical value. These values are described in subsequent sections. Eighteen (18) bytes are reserved in the header for future expansion.

-23-

A type/subtype specific data area follows the header. This area can be up to four hundred and forty-eight (448) bytes in length.

Message Channel

- The LAS is provided with a message channel (MC) with a signaling rate of 5 2.048 Mbps to provide a communications channel between RM channel units and the SM. The idle condition of the MC is defined as all 1s. Bus contention is detected when a module writes a 1 onto the MC, but simultaneously reads a 0. Bus contention is resolved as follows:
- 1) The module detecting contention shall immediately abort its message
10 frame and go into a high impedance state.
- 2) The module shall not attempt retransmission until it detects that the MC is in the idle state.
- 3) Once a full frame has been transmitted the module shall not attempt to transmit another frame until it has detected 10 successive 1s on the MC.
15 All messages, as shown in the message diagram below at Table 1, whether a request-response type or an autonomous type are constrained to be of one common form. This standardizes message validation in the software, and also allows future message types to be more easily added.

Message Administration

- 20 This information allows the message to be recognized, to be uniquely identified, and to describe an operation to be performed or information to be passed.

Addressing

Origination

- 25 Uniquely describes the entity creating the message. It contains both physical and logical information.

Data

This section of the message contains data which is specific to the message type. Some examples of this are: provisioning information, and downloaded software.

Table 1

A header is at the beginning of each message. Fields in the header are shared by all of the message types. These fields are validated by the software for each received message. If a header field contains illogical information, an attempt is made to inform the sender of the message of the problem. If the header is sufficiently corrupt, this may be impossible. The message is then discarded by the receiving entity. The message originating entity is tasked with ensuring that a message that requires a response actually gets one. This can be in the form of an acknowledgment or a time-out. The message originator is ultimately responsible for the message until it is acknowledged. If the originator deems it appropriate, the message can be sent again, or some other action can be taken.

-25-

The message domain field in the header is used to characterize the originating and destination control domain of a message.

- A carbon copy message (CC) indication is included in this field. These copy messages are created and sent for information purposes only, and are never acknowledged, regardless of the message type. These messages could notify some other “non-involved” entity that some message transaction has taken place. For example, a RM to RM transaction may send a carbon copy to the SM for logging and information purposes. This CC message sending ability is able to be turned on and off via configuration parameters in the RMs and SMs.

10 The following message domains are available.

D_PROC - (0x01)

This domain is used for messages that remain within a SM. Inter-process messages would use this domain. The value 0x01 is placed in the message domain field. The message type varies, and more specifically characterizes the message.

15 D_PROC_CC - (0x81)

This domain is used for carbon copy messages that remain within a SM. Inter-process messages would use this domain. The value 0x81 is placed in the message domain field. The message type varies, and more specifically characterizes the message.

20 D_INTRA - (0x02)

This domain is used for messages that remain within a physical shelf. Local SM or RM communications would use this domain. The value 0x02 is placed in the message domain field. The message type varies, and more specifically characterizes the message.

25 D_INTRA_CC - (0x82)

This domain is used for carbon copy messages that remain within a physical shelf. Local SM to RM communications would use this domain. The value 0x82 is placed in the message domain field. The message type varies, and more specifically characterizes the message.

-26-

D_INTER - (0x03)

This domain is used for messages that are destined for a different physical shelf than that in which the originating entity resides, but within the same node. The value 0x03 is placed in the message domain field. The message type varies, and more 5 specifically characterizes the message.

D_INTER_CC - (0x83)

This domain is used for carbon copy messages that are destined for a different physical shelf than that in which the originating entity resides, but within the same node. The value 0x83 is placed in the message domain field. The message type varies, 10 and more specifically characterizes the message.

D_NODAL - (0x04)

This domain is used for messages that are destined for a different logical node than that in which the originating entity resides. The value 0x04 is placed in the message domain field. The message type varies, and more specifically characterizes the 15 message.

D_NODAL_CC - (0x84)

This domain is used for carbon copy messages that are destined for a different logical node than that in which the originating entity resides. The value 0x84 is placed in the message domain field. The message type varies, and more specifically 20 characterizes the message.

D_DOMAIN - (0x05)

This domain is used for messages that are destined for a different physical domain than that in which the originating entity resides. The value 0x05 is placed in the message domain field. The message type varies, and more specifically characterizes the 25 message.

D_DOMAIN_CC - (0x85)

This domain is used for carbon copy messages that are destined for a different physical domain than that in which the originating entity resides. The value 0x85 is placed in the message domain field. The message type varies, and more specifically 30 characterizes the message.

-27-

The message type and message subtype fields are used in conjunction to characterize the functionality of a message. The following message types are available.

MSG_A_INFO - (0x10)

This message type is used for a message that requires an acknowledgment. The 5 value 0x10 is placed in the message type field. The message subtype varies, and more specifically characterizes the message. It is typically used to inform another entity of an event. For example, a RM that wants to inform the SM of a significant event will use this message type. MSG_A_INFO messages are among the most commonly used of all the types.

10 MSG_U_INFO - (0x20)

This message type is used for a message that does not require an acknowledgment. The value 0x20 is placed in the message type field. The message subtype varies, and more specifically characterizes the message. This type is used for non-critical, "information only" messages. A keep-alive message might fall into this 15 category.

MSG_GET - (0x30)

This message type is used to get information from the message recipient. The value 0x30 is placed in the message type field. The message subtype varies. This is used to retrieve the value(s) of variable(s) controlled by the message recipient. As an 20 example, a SM would request the software version of a RM with this message type. MSG_GET messages are always acknowledged. The acknowledgment should contain the requested information in the Data portion, and the Status field will reflect the success or failure of the operation. If successful, a MSG_GET has returned ALL of the requested variables. If unsuccessful, NONE of the requested variables are returned.

25 MSG_SET - (0x40)

This message type is used to impart information to the message recipient. The value 0x40 is placed in the message type field. The message subtype varies. This is used to MSG_SET the value(s) of variable(s) controlled by the message recipient. As an example, a SM would send provisioning information to be saved in a RM via this 30 type. MSG_SET messages are always acknowledged. The acknowledgment should reflect the success or failure of the operation via the Status field. If successful, a

-28-

MSG_SET has configured ALL of the requested variables. If unsuccessful, NONE of the requested variables are configured.

MSG_A_ACTION - (0x50)

This message type is used for a message that requires an acknowledgment. The value 0x50 is placed in the message type field. The message subtype varies, and more specifically characterizes the message. It is typically used to request an action be performed by another entity. For example, a SM that wants to have a RM reset itself will use this message type. MSG_A_ACTION messages are among the most commonly used of all the types.

10 **MSG_ACK - (0x60)**

This message type is used to acknowledge the receipt of a message. The value 0x60 is placed in the message type field. The message subtype varies. This type is used to acknowledge a previous MSG_A_ACTION, MSG_A_INFO, MSG_GET or MSG_SET message. The subtype used will be the same as the received message subtype. The Status field is critical to and required by this message type. There is no response to an MSG_ACK message.

Message Subtype

S_RM_ILLOG - (0x00)

The S_RM_ILLOG subtype is used by the SM and the RM for debugging purposes only. It is not to be used under normal circumstances. The responding RM or SM acknowledges with this subtype. The value 0x00 is placed in the message subtype field. The message type varies, and more specifically characterizes the message. The request uses any type, and the response uses the MSG_ACK type.

S_RM_ID - (0x01)

25 The S_RM_ID subtype is used by the SM to request that a RM respond with its identity. The receiving RM acknowledges with this subtype along with the appropriate identifying information in the data field. This information will consist of the board identification and the software version identification. The value 0x01 is placed in the message subtype field. The message type varies, and more specifically characterizes the

-29-

message. The message type varies, and more specifically characterizes the message. The request uses the MSG_GET type, and the response uses the MSG_ACK type.

S_RM_DIAG - (0x02)

The S_RM_DIAG subtype is used by the SM to request that a RM perform a self-diagnostic test and respond with the results of the test. The receiving RM acknowledges with this subtype along with the appropriate diagnostic results information in the data field. The value 0x02 is placed in the message subtype field. The message type varies, and more specifically characterizes the message. The request uses the MSG_A_ACTION type, and the response uses the MSG_ACK type.

10 S_RM_SW - (0x03)

The S_RM_SW subtype is used by the SM to implement a software download to a RM. The RM responds by acknowledging the receipt and storage of the downloaded software segment contained in the message. This software will be contained in the data portion of the message. Multiple S_RM_SW messages will be sent until all of the software is on the target RM. The value 0x03 is placed in the message subtype field. The message type varies, and more specifically characterizes the message. The request uses the MSG_A_ACTION type, and the response uses the MSG_ACK type. Since the downloaded software is contained in multiple messages, the Segment number field is also used.

20 S_RM_PROV - (0x04)

The S_RM_PROV subtype is used by the SM to transfer provisioning and configuration information to or from a RM. The RM responds by acknowledging the receipt and storage of the information segment contained in the message to transfers the current values of its provisioning to the SM. This information will be contained in the data portion of the message. Multiple S_RM_PROV messages will be sent until the transfer is complete. The value 0x04 is placed in the message subtype field. The message type varies, and more specifically characterizes the message. The request uses the MSG_SET type or the MSG_GET type, and the response uses the MSG_ACK type. If the provisioning and configuration are contained in multiple messages, the Segment number field is also used.

30 S_RM_READY - (0x05)

-30-

- This S_RM_READY subtype is used by the SM to request that a RM inform the SM when it has begun online operation. The receiving RM acknowledges with this subtype when it has completely initialized itself, and is attached to the transmission facilities. The value 0x05 is placed in the message subtype field. The message type 5 varies, and more specifically characterizes the message. The request uses the MSG_A_ACTION type, and the response uses the MSG_ACK type.

S_RM_RTU - (0x06)

- The S_RM_RTU subtype is used by the SM or a specialized RM to initiate a test on a target RM. The target RM responds by acknowledging the receipt of the message 10 and the completion, or progress, of the requested test operation. If the progress of a test requires multiple MSG_ACK messages, the Message Segment field will also be used. The test results will be contained in the data portion of the MSG_ACK message. The value 0x06 is placed in the message subtype field. The message type varies, and more specifically characterizes the message. The request uses the MSG_A_ACTION type, 15 and the response uses the MSG_ACK type. The Message Operation field typically specifies the type of test to be performed.

S_RM_RESET - (0x07)

- This S_RM_RESET subtype is used by the SM to request that a RM reset or reinitialize itself. The receiving RM cleans up, removes itself from the bus and resets 20 itself. The value 0x07 is placed in the message subtype field. The request uses the MSG_U_INFO type, and there is no response expected. The SM would expect to eventually receive a S_RM_INSERTED message from the RM.

S_RM_INSERTED - (0x08)

- This S_RM_INSERTED subtype is used by the RM to inform the SM that it has 25 just been inserted or initialized, and is ready to be configured. The value 0x08 is placed in the message subtype field. The request uses the MSG_U_INFO type, and there is no response expected. The RM would eventually receive an S_RM_ID request from the SM. A RM would periodically resend S_RM_INSERTED messages until it sees the S_RM_ID message.

- 30 S_RM_GO_OFFLINE - (0x09)

-31-

This S_RM_GO_OFFLINE subtype is used by the SM to request that a RM go into an offline, non-transmission state. The receiving RM acknowledges with this subtype when it has removed itself from the bus, and is ready to be reconnected. The value 0x09 is placed in the message subtype field. The message type varies, and more specifically characterizes the message. The request uses the MSG_A_ACTION type, and the response uses the MSG_ACK type.

S_RM_SIG_EVENT - (0x0a)

This S_RM_SIG_EVENT subtype is used by the RM to inform the SM that it has detected a reportable event. The SM acknowledges with this subtype when it has logged the event. The event information (event, alarm, performance information, etc.) will be contained in the data portion of the MSG_A_INFO message. The value 0x0a is placed in the message subtype field. The message type varies, and more specifically characterizes the message. The request uses the MSG_A_INFO type, and the response uses the MSG_ACK type.

15 S_RM_INFO - (0x0b)

This S_RM_INFO subtype is used by the RM to inform the SM that it has detected something that may be of interest to the SM, but is not necessarily reportable. The SM acknowledges with this subtype when it noted the information. The information will be contained in the data portion of the MSG_A_INFO or MSG_SET message. The value 0x0b is placed in the message subtype field. The message type varies, and more specifically characterizes the message. The request uses the MSG_A_INFO and MSG_SET types, and the response uses the MSG_ACK type.

S_SM_ILLOG - (0x0c)

The S_SM_ILLOG subtype is used by the SM for debugging purposes only. It is not to be used under normal circumstances. The responding SM acknowledges with this subtype. The value 0x0c is placed in the message subtype field. The message type varies, and more specifically characterizes the message. The request uses any type, and the response uses the MSG_ACK type.

S_SM_ID - (0x0d)

30 The S_SM_ID subtype is used by the master SM to request that another SM respond with its identity. The receiving SM acknowledges with this subtype along with

-32-

the appropriate identifying information in the data field. This information will consist of the board identification, shelf number, SM unique identifier, and software version identification. The value 0x0d is placed in the message subtype field. The message type varies, and more specifically characterizes the message. The request uses the

- 5 MSG_GET type, and the response uses the MSG_ACK type.

S_SM_DIAG - (0x0e)

The S_SM_DIAG subtype is used by the master SM to request that another SM perform a self-diagnostic test and respond with the results of the test. The receiving SM acknowledges with this subtype along with the appropriate diagnostic results

- 10 information in the data field. The value 0x0e is placed in the message subtype field. The message type varies, and more specifically characterizes the message. The request uses the MSG_A_ACTION type, and the response uses the MSG_ACK type.

S_SM_SW - (0x0f)

- The S_SM_SW subtype is used by the master SM to implement a software download to another SM. The SM responds by acknowledging the receipt and storage of the downloaded software segment contained in the message. This software will be contained in the data portion of the message. Multiple S_SM_SW messages will be sent until all of the software is on the target SM. The value 0x0f is placed in the message subtype field. The message type varies, and more specifically characterizes the message. The request uses the MSG_A_Action type, and the response uses the MSG_ACK type. Since the downloaded software is contained in multiple messages, the Segment number field is also used.

S_SM_PROV - (0x10)

- The S_SM_PROV subtype is used by the master SM to transfer provisioning and configuration information to or from a SM. The SM responds by acknowledging the receipt and storage of the information segment contained in the message or transfers the current values of its provisioning to the master SM. This information will be contained in the data portion of the message. Multiple S_SM_PROV messages will be sent until the transfer is complete. The value 0x10 is placed in the message subtype field. The message type varies, and more specifically characterizes the message. The request uses the MSG_SET type or the MSG_GET type, and the response uses the

-33-

MSG_ACK type. If the provisioning and configuration are contained in multiple messages, the Segment number field is also used.

S_SM_READY - (0x11)

This S_SM_READY subtype is used by the master SM to request that another

- 5 SM inform the master SM when it has begun online operation. The receiving SM acknowledges with this subtype when it has completely initialized itself. The value 0x11 is placed in the message subtype field. The message type varies, and more specifically characterizes the message. The request uses the MSG_A_ACTION type, and the response uses the MSG_ACK type.

10 S_SM_RTU - (0x12)

The S_SM_RTU subtype is used by the master SM or a specialized RM to initiate a test on a target SM. The target SM responds by acknowledging the receipt of the message and the completion, or progress, of the requested test operation. If the progress of a test requires multiple MSG_ACK messages, the Message Segment field 15 will also be used. The test results will be contained in the data portion of the MSG_ACK message. The value 0x12 is placed in the message subtype field. The message type varies, and more specifically characterizes the message. The request uses the MSG_A_ACTION type, and the response uses the MSG_ACK type. The Message Operation field typically specifies the type of test to be performed.

20 S_SM_RESET - (0x13)

This S_SM_RESET subtype is used by the master SM to request that another SM reset or reinitialize itself. The receiving SM cleans up, and resets itself. The value 0x13 is placed in the message subtype field. The request uses the MSG_U_INFO type, and there is no response expected. The master SM would expect to eventually receive a 25 S_SM_INSERTED message from the SM.

S_SM_INSERTED - (0x14)

This S_SM_INSERTED subtype is used by the SM to inform the master SM that it has just been inserted or initialized, and is ready to be configured. The value 0x14 is placed in the message subtype field. The request uses the MSG_U_INFO type, 30 and there is no response expected. The SM would eventually receive an S_SN_ID

-34-

request from the SM. A SM would periodically resend S_SM_INSERTED messages until it sees the S_SM_ID message.

S_SM_GO_OFFLINE (0x15)

- This S_SM_GO_OFFLINE subtype is used by the master SM to request that a SM go into an offline state. The receiving SM acknowledges with this subtype when it has removed itself from online operation, and is ready to be reconnected. The value 0x15 is placed in the message subtype field. The message type varies, and more specifically characterizes the message. The request uses the MSG_A_ACTION type, and the response uses the MSG_ACK type.

10 S_SM_SIG_EVENT - (0x16)

This S_SM_SIG_EVENT subtype is used by the SM to inform the master SM that it has detected a reportable event. The master SM acknowledges with this subtype when it has logged the event. The event information (event, alarm, performance information, etc.) will be contained in the data portion of the MSG_A_INFO message.

- 15 The value 0x16 is placed in the message subtype field. The message type varies, and more specifically characterizes the message. The request uses the MSG_A_INFO type, and the response uses the MSG_ACK type.

- S_SM_INFO subtype is used by the SM to inform the master SM that it has detected something that may be of interest to the SM, but is not necessarily reportable.
- 20 The master SM acknowledges with this subtype when it noted the information. The S_SM_INFO is also used by the master SM to report information of interest to another SM. The SM acknowledges with this subtype when it noted the information. The information will be contained in the data portion of the MSG_A_INFO or MSG_SET message. The value 0x17 is placed in the message subtype field. The message type varies, and more specifically characterizes the message. The request uses the MSG_A_INFO and MSG_SET types, and the response uses the MSG_ACK type.

S_SM_STATE - (0x18)

- The S_SM_STATE subtype is used by the master SM to transfer operational state information to or from a SM. The SM responds by acknowledging the receipt and storage of the information segment contained in the message or transfers the current values of its state information to the master SM. This information will be contained in

-35-

the data portion of the message. Multiple S_SM_STATE messages will be sent until the transfer is complete. The value 0x18 is placed in the message subtype field. The message type varies, and more specifically characterizes the message. The request uses the MSG_SET type or the MSG_GET type. If the information is contained in multiple 5 messages, the Segment number field is also used.

S_RM_STATE - (0x19)

The S_RM_STATE subtype is used by the SM to transfer operational state information to or from a RM. the RM responds by acknowledging the receipt and storage of the information segment contained in the message or transfers the current 10 values of its state information to the SM. This information will be contained in the data portion of the message. Multiple S_RM_STATE messages will be sent until the transfer is complete. The value 0x19 is placed in the message subtype field. The message type varies, and more specifically characterizes the message. The request uses the MSG_SET type or the MSG_GET type, and the response uses the MSG_ACK type. 15 If the information is contained in multiple messages, the Segment number field is also used.

S_SM_TIMER - (0x20)

The S_SM_TIMER subtype is used by the SM to time message transfers. The message header for the message to be timed will be stored in the data portion of the 20 message, along with the timer value. The value 0x20 is placed in the message subtype field. The message type varies, and more specifically characterizes the message. The request uses the MSG_U_INFO type.

S_RM_TIMER - (0x21)

The S_RM_TIMER subtype is used by the RM to time message transfers. The 25 message header for the message to be timed will be stored in the data portion of the message, along with the timer value. The value 0x21 is placed int he message subtype field. The message type varies, and more specifically characterizes the message. The request uses the MSG_U_INFO type.

S_NM_TIMER - (0x22)

30 The S_NM_TIMER subtype is used by the NM to time message transfers. The message header for the message to be timed will be stored in the data portion of the

-36-

message, along with the timer value. The value 0x22 is placed in the message subtype field. The message type varies, and more specifically characterizes the message. The request uses the MSG_U_INFO type.

S_RM_CONN_TS - (0x23)

- 5 The S_RM_CONN_TS subtype is used by the SM to transfer timeslot cross connection information to or from a RM. The RM responds by acknowledging the receipt and storage of the information segment contained in the message or transfers the current values of its timeslot connection information to the SM. This information will be contained in the data portion of the message. Multiple S_RM_CONN_TS messages
10 will be sent until the transfer is complete. The value 0x23 is placed in the message subtype field. The message type varies, and more specifically characterizes the message. The request uses the MSG_SET type or the MSG_GET type, and the response uses the MSG_ACK type. If information is contained in multiple messages, the Segment number field is also used.

15 Message Operation

OP_DUMMY - (0x00)

- The OP_DUMMY operation is used by the SM and the RM as a placeholder for message subtypes that do not require an operation type. It is not to be used under any other circumstances. The responding RM or SM acknowledges with the OP_DUMMY operation type. The value 0x00 is placed in the message operation field. The message subtype varies, and more specifically characterizes the message. When the request uses the MSG_A_ACTION, MSG_A_INFO, MSG_GET or MSG_SET type, the response uses the MSG_ACK type. When the request uses the MSG_U_INFO type, there is no response expected.

25 OP_SW_RM - (0x01)

- This OP_SW_RM operation is used by the SM to inform the RM that the message contains software for the RM to save. The RM acknowledges with this operation type after it saves the information contained in the message data field. The downloaded software will be contained in the data portion of the MSG_A_ACTION message. The value 0x01 is placed in the message operation field. The message

-37-

subtype is S_RM_SW, and more specifically characterizes the message. The request uses the MSG_A_ACTION type, and the response uses the MSG_ACK type.

OP_SW_RM_FINAL (0x02)

This OP_SW_RM_FINAL operation is used by the SM to inform the RM that it has received the final pieces of the downloaded software. The RM acknowledges with this operation type when saves the software. The final downloaded software piece will be contained in the data portion of the MSG_A_ACTION message. The value 0x02 is placed in the message operation field. The Message Segment field will contain the next sequential number in the sequence generated in the OP_SW_RM messages. The message type is S_RM_SW, and more specifically characterizes the message. The request uses the MSG_A_ACTION type, and the response uses the MSG_ACK type.

OP_TEST_MON_ENABLE_DIG - (0x03)

This OP_TEST_MON_ENABLE_DIG operation is used by the SM to request that the RM prepare itself to be tested digitally in monitor mode. This message is sent at the initiation of a testing session. The RM acknowledges with this operation type when it is prepared for further testing operations. There is no contained in the data portion of the MSG_A_ACTION message. The value 0x03 is placed in the message operation field. The message subtype is S_RM_RTU, and more specifically characterizes the message. The request uses the MSG_A_ACTION type, and the response uses the MSG_ACK type.

OP_TEST_SPLT_ENABLE_DIG - (0x04)

This OP_TEST_SPLT_ENABLE_DIG operation is used by the SM to request that the RM prepare itself to be tested digitally in split mode. This message is sent at the initiation of a testing session. The RM acknowledges with this operation type when it is prepared for further testing operations. The data portion of the MSG_A_ACTION message contains the “direction” of the split. The value 0x04 is placed in the message operation field. The message subtype is S_RM_RTU, and more specifically characterizes the message. The request uses the MSG_A_ACTION type, and the response uses the MSG_ACK type.

30 OP_TEST_MON_ENABLE_MET (0x05)

-38-

- This OP_TEST_MON_ENABLE_MET operation is used by the SM to request that the RM prepare itself for metallic testing in monitor mode. This message is sent at the initiation of a testing session. The RM acknowledges with this operation type when it is prepared for further testing operations. There is no contained in the data portion of
- 5 the MSG_A_ACTION message. the value 0x05 is placed in the message operation field. The message subtype is S_RM_RTU, and more specifically characterizes the message. The request uses the MSG_A_ACTION type, and the response uses the MSG_ACK type.

OP_TEST_SPLT_ENABLE_MET - (0x06)

- 10 This OP_TEST_SPLT_ENABLE_MET operation is used by the SM to request that the RM prepare itself for metallic testing in split mode. This message is sent at the initiation of a testing session. The RM acknowledges with this operation type when it is prepared for further testing operations. The data portion of the MSG_A_ACTION message contains the “direction” of the split. The value 0x06 is placed in the message
- 15 operation field. The message subtype is S_RM_RTU, and more specifically characterizes the message. The request uses the MSG_A_ACTION type, and the response uses the MSG_ACK type.

OP_TEST_MODE_DISABLE - (0x07)

- This OP_TEST_MODE_DISABLE operation is used by the SM to request that
- 20 the RM remove itself from test mode. This message is sent at the termination of a testing session, and is effective for all test Enable types. The RM acknowledges with this operation type when it is no longer in test mode, and has resumed normal operation. There is no contained in the data portion of the MSG_A_ACTION message. The value 0x07 is placed in the message operation field. The message subtype is S_RM_RTU,
- 25 and more specifically characterizes the message. The request uses the MSG_A_ACTION type, and the response uses the MSG_ACK type.

OP_TEST_LOOP - (0x08)

- This OP_TEST_LOOP operation is used by the SM to request that the RM perform a loopback. The RM acknowledges with this operation type when it has
- 30 completed the requested test operation. There is no contained in the data portion of the MSG_A_ACTION message. The value 0x08 is placed in the message operation field.

-39-

The message subtype is S_RM_RTU, and more specifically characterizes the message. The request uses the MSG_A_ACTION type, and the response uses the MSG_ACK type.

OP_TEST_UNLOOP - (0x09)

- 5 This OP_TEST_UNLOOP operation is used by the SM to request that the RM release a loopback. The RM acknowledges with this operation type when it has completed the requested test operation. There is no contained in the data portion of the MSG_A_ACTION message. The value 0x09 is placed in the message operation field. The message subtype is S_RM_RTU, and more specifically characterizes the message.
- 10 The request uses the MSG_A_ACTION type, and the response uses the MSG_ACK type.

OP_EVENT_ALARM - (0x0a)

- This OP_EVENT_CRITICAL operation is used by the RM to inform the SM, or the SM to inform the master SM of an alarm. The SM acknowledges with this 15 operation type when it has completed logged the reported event. The data portion of the MSG_A_INFO message contains the specific event to be reported. The value 0x0a is placed in the message operation field. The message subtype is S_RM_SIG_EVENT (RM originated) or S_SM_SIG_EVENT (SM originated), and more specifically characterizes the message. The request uses the MSG_A_INFO type, and the response 20 uses the MSG_ACK type.

OP_EVENT_GEN - (0x0d)

- This OP_EVENT_GEN operation is used by the RM to inform the SM, or the SM to inform the master SM of a reportable event. The SM acknowledges with this 25 operation type when it has completed logged the reported event. The data portion of the MSG_A_INFO message contains the specific event to be reported. The value 0x0d is placed in the message field. The message subtype is S_RM_SIG_EVENT (RM originated) or S_SM_SIG_EVENT (SM originated), and more specifically characterizes the message. The request uses the MSG_A_INFO type, and the response uses the MSG_ACK type.

- 30 OP_EVENT_PERF - (0x0e)

-40-

This OP_EVENT_PERF operation is used by the RM to inform the SM, or the SM to inform the master SM of a reportable performance monitoring event. The SM acknowledges with this operation type when it has completed logged the reported event. The data portion of the MSG_A_INFO message contains the specific event to be 5 reported. The value 0x0e is placed in the message operation field. The message subtype is S_RM_SIG_EVENT (RM originated) or S_SM_SIG_EVENT (SM originated), and more specifically characterizes the message. The request uses the MSG_A_INFO type, and the response uses the MSG_ACK type.

OP_SW_RM_SWITCH - (0x0f)

10 This OP_SW_RM_SWITCH operation is used by the SM to inform the RM it should begin executing the newest downloaded software. The RM acknowledges with this operation type when has prepared to perform the operation. The data portion of the MSG_A_ACTION message contains the version number of the previously downloaded software. The RM will reset itself as a result of receiving this message. The value 0x0f 15 is placed in the message operation field. The message type is S_RM_SW, and more specifically characterizes the message. The request uses the MSG_A_ACTION type, and the response uses the MSG_ACK type. The RM will subsequently reset itself and execute the current software.

OP_PROV_RM_FACIL - (0x10)

20 This OP_PROV_RM_FACIL operation is used by the SM to transfer facility provisioning and configuration information to or from a RM. The RM responds by acknowledging the receipt and storage of the information segment contained in the message or transfers the current values of its provisioning to the SM. This facility information will be contained in the data portion of the message. Multiple 25 OP_PROV_RM_FACIL messages will be sent until the transfer is complete. The value 0x10 is placed in the message operation field. The message subtype is S_RM_PROV, and more specifically characterizes the message. The request uses the MSG_SET type or the MSG_GET type, and the response uses the MSG_ACK type. If the provisioning and configuration are contained in multiple messages, the Segment number field is also 30 used.

OP_PROV_RM_SYS - (0x11)

-41-

This OP_PROV_RM_SYS operation is used by the SM to transfer system parameter configuration information to or from a RM. These could include shelf identifying data, timer values to be used, etc. The RM responds by acknowledging the receipt and storage of the information segment contained in the message or transfers the 5 current values of its provisioning to the SM. This information will be contained in the data portion of the message. Multiple OP_PROV_RM_SYS messages will be sent until the transfer is complete. The value 0x11 is placed in the message operation field. The message subtype is S_RM_PROV, and more specifically characterizes the message. The request uses the MSG_SET type or the MSG_GET type, and the response uses the 10 MSG_ACK type. If the provisioning and configuration are contained in multiple messages, the Segment number field is also used.

OP_SW_SM - (0x12)

This OP_SW_SM operation is used by the master SM to inform the SM that the message contains software for the SM to save. The SM acknowledges with this 15 operation type after it saves the information contained in the message data field. The downloaded software will be contained in the data portion of the MSG_A_ACTION message. The value 0x12 is placed in the message operation field. The message subtype is S_SM_SW, and more specifically characterizes the message. The request uses the MSG_A_ACTION type, and the response uses the MSG_ACK type.

20 OP_SW_SM_FINAL - (0x13)

This OP_SW_SM_FINAL operation is used by the master SM that it has received the final pieces of the downloaded software. The SM acknowledges with this operation type when saves the software. The final downloaded software piece will be contained in the data portion of the MSG_A_ACTION message. The value 0x13 is 25 placed in the message operation field. The Message Segment field will contain the next sequential number in the sequence generated in the OP_SW_SM messages. The message type is S_SM_SW, and more specifically characterizes the message. The request uses the MSG_A_ACTION type and the response uses the MSG_ACK type.

OP_SW_SM_SWITCH - (0x14)

30 This OP_SW_SM_SWITCH operation is used by the master SM to inform the SM it should begin executing the newest downloaded software. The SM acknowledges with this operation type when has prepared to perform the operation. The data portion

-42-

- of the MSG_A_ACTION message contains the version number of the previously downloaded software. The SM will reset itself as a result of receiving this message. The value 0x14 is placed in the message operation field. The message type is S_SM_SW, and more specifically characterizes the message. The request uses the
- 5 MSG_A_ACTION type, and the response uses the MSG_ACK type. The RM will subsequently reset itself and execute the current software.

OP_PROV_SM_SYS - (0x15)

- This OP_PROV_SM_SYS operation is used by the master SM to transfer system parameter configuration information to or from a SM. These could include shelf
- 10 identifying data, timer values to be used, etc. The SM responds by acknowledging the receipt and storage of the information segment contained in the message or transfers the current values of its provisioning to the master SM. This information will be contained in the data portion of the message. Multiple OP_PROV_SYS messages will be sent until the transfer is complete. The value 0x15 is placed in the message operation field.
- 15 The message subtype is S_SM_PROV, and more specifically characterizes the message. The request uses the MSG_SET type or the MSG_GET type, and the response uses the MSG_ACK type. If the provisioning and configuration are contained in multiple messages, the Segment number field is also used.

OP_STATE_RM_FACIL - (0x16)

- 20 This OP_STATE_RM_FACIL operation is used by the SM to transfer facility state information to or from a RM. The RM responds by acknowledging the receipt and storage of the information segment contained in the message or transfers the current values of its state to the SM. This facility information will be contained in the data portion of the message. Multiple OP_STATE_RM_FACIL messages will be sent until
- 25 the transfer is complete. The value 0x16 is placed in the message operation field. The message subtype is S_RM_STATE, and the response uses the MSG_ACK type. If the provisioning and configuration are contained in multiple messages, the Segment number field is also used.

OP_STATE_RM_SYS - (0x17)

- 30 This OP_STATE_RM_SYS operation is used by the SM to transfer RM state information to or from a RM. The RM responds by acknowledging the receipt and

-43-

storage of the information segment contained in the message or transfers the current values of its provisioning to the SM. This information will be contained in the data portion of the message. Multiple OP_STATE_RM_SYS messages will be sent until the transfer is complete. The value 0x17 is placed in the message operation field. The 5 message subtype is S_RM_STATE, and more specifically characterizes the message. The request uses the MSG_SET type or the MSG_GET type, and the response uses the MSG_ACK type. If the information is contained in multiple messages, the Segment number field is also used.

OP_STATE_SM_SYS - (0x18)

10 This OP_PROV_SM_SYS operation is used by the master SM to transfer state information to or from a SM. The SM responds by acknowledging the receipt and storage of the information segment contained in the message or transfers the current values of its information to the master SM. This information will be contained in the data portion of the message; Multiple OP_STATE_SM_SYS messages will be sent 15 until the transfer is complete. The value 0x18 is placed in the message operation field. The message subtype is S_SM_STATE, and more specifically characterizes the message. The request uses the MSG_SET type or the MSG_GET type, and the response uses the MSG_ACK type. If the information is contained in multiple messages, the Segment number field is also used.

20 **OP_SYNC_TICK_SM - (0x19)**

This OP_SYNC_TICK_SM operation is used by the master SM to cause a time tick re-synchronization to take place in a SM. The SM responds by acknowledging the receipt of the information contained in the message and resets his tick sync to the value of the information received from the master SM. This information will be contained in 25 the data portion of the message. The value 0x19 is placed in the message operation field. The message subtype is S_SM_INFO, and more specifically characterizes the message. The request uses the MSG_SET type, and the response uses the MSG_ACK type.

OP_SYNC_TICK_RM - (0x20)

30 This OP_SYNC_TICK_SM operation is used by the SM to cause a time tick re-synchronization to take place in a RM. The RM responds by acknowledging the receipt

-44-

of the information contained in the message and resets his tick sync to the value of the information received from the SM. This information will be contained in the data portion of the message. The value 0x20 is placed in the message operation field. The message subtype is S_RM_INFO, and more specifically characterizes the message. The 5 request uses the MSG_SET type, and the response uses the MSG_ACK type.

OP_SYNC_TIME_SM - (0x21)

This OP_SYNC_TIME_SM operation is used by the master SM or a NM to cause a time re-synchronization to take place in a SM or master SM. The SM responds by acknowledging the receipt of the information contained in the message and resets his 10 time to the value of the information received from the master SM or NM. This information will be contained in the data portion of the message. The value 0x21 is placed in the message operation field. The message subtype is S_SM_INFO, and more specifically characterizes the message. The request uses the MSG_SET type, and the response uses the MSG_ACK type.

OP_WATCHDOG_RM - (0x22)

This OP_WATCHDOG_RM operation is used by the RM to inform the SM that the RM is still alive. This message is sent on a periodic basis if no other message activity has recently transpired. The data portion of the MSG_A_INFO message is unused. The value 0x22 is placed in the message operation field. The message subtype 20 is S_RM_INFO and more specifically characterizes the message. The request uses the MSG_U_INFO type, and there is no response.

OP_WATCHDOG_SM - (0x23)

This OP_WATCHDOG_SM operation is used by the SM to inform the master SM that the SM is still alive. This message is sent on a periodic basis if no other 25 message activity has recently transpired. The data portion of the MSG_A_INFO message is unused. The value 0x23 is placed in the message operation field. The message subtype is S_SM_INFO and more specifically characterizes the message. The request uses the MSG_U_INFO type, and there is no response.

OP_SW_SAV_RM - (0x24)

30 This OP_SW_SAV_RM operation is used by the NM or master SM to inform the SM that the message contains software for a RM that is to be saved on the SM

-45-

- storage media. The SM acknowledges with this operation type after it saves the information contained in the message data field. The software will be contained in the data portion of the MSG_A_ACTION message. The value 0x24 is placed in the message operation field. The message subtype is S_SM_SW, and more specifically characterizes the message. The request uses the MSG_A_ACTION type, and the response uses the MSG_ACK type.
- 5

OP_SW_SAV_RM_FINAL - (0x25)

- This OP_SW_SAV_RM_FINAL operation is used by the NM or master SM to inform the SM that the final message containing software for a RM that is to be saved on the SMs storage media has been sent. The SM acknowledges with this operation type. There is nothing contained in the data portion of the MSG_A_ACTION message. The value 0x25 is placed in the message operation field. The message subtype is S_SM_SW, and more specifically characterizes the message. The request uses the MSG_A_ACTION type, and the response uses the MSG_ACK type.
- 10

15 OP_SW_SAV_SM - (0x26)

- This OP_SW_SAV_RM operation is used by the NM or master SM to inform the SM that the message contains software for a SM that is to be saved on the SMs storage media. The SM acknowledges with this operation type after it saves the information contained in the message data field. The software will be contained in the data portion of the MSG_A_ACTION message. The value 0x26 is placed in the message operation field. The message subtype is S_SM_SW, and more specifically characterizes the message. The request uses the MSG_A_ACTION type, and the response uses the MSG_ACK type.
- 20

OP_SW_SAV_SM_FINAL - (0x27)

- 25 This OP_SW_SAV_SM_FINAL operation is used by the NM or master SM to inform the SM that the final message containing software for AM that is to be saved on the SMs storage media has been sent. The SM acknowledges with this operation type. There is nothing contained in the data portion of the MSG_A_ACTION message. The value 0x27 is placed in the message operation field. The message subtype is S_SM_SW, and more specifically characterizes the message. The request uses the MSG_A_ACTION type, and the response uses the MSG_ACK type.
- 30

-46-

OP_PROV_SAV_RM - (0x28)

- This OP_SW_SAV_RM operation is used by the NM or master SM to transfer facility provisioning and configuration information to or from a SM for a RM that is to be saved on the SMs storage media. The SM acknowledges with this operation type
- 5 after it saves the information contained in the data field of the MSG_SET message. The data portion will also identify the RM information being queried. The provisioning and configuration information will be contained in the data portion of the acknowledgment of the MSG_GET message. The value 0x28 is placed in the message operation field. The message subtype is S_SM_PROV, and more specifically characterizes the message.
- 10 The request uses the MSG_GET or the MSG_SET type and the response uses the MSG_ACK type.

OP_TS_CONN_SAV_RM (0x29)

- This OP_TS_CONN_SAV_RM operation is used by the NM or master SM to transfer timeslot cross connection information to or from a SM for a RM that is to be saved on the SMs storage media. The SM acknowledges with this operation type after it saves the information contained in the data field of the MSG_SET message. The data portion will also identify the RM information being queried. The timeslot cross connection information will be contained in the data portion of the acknowledgment of the MSG_GET message. The value 0x29 is placed in the message operation field. The message subtype is S_RM_CONN_TS, and more specifically characterizes the message. The request uses the MSG_GET or the MSG_SET type, and the response uses the MSG_ACK type.

OP_TS_CONN_RM - (0x2a)

- This OP_TS_CONN_RM operation is used by the SM to transfer timeslot cross connection information to or from a RM. The RM acknowledges with this operation type after it saves the information contained in the data field of the MSG_SET message. The timeslot cross connection information will be contained in the data portion of the acknowledgment of the MSG_GET message. The value 0x2a is placed in the message operation field. The message subtype is S_RM_CONN_TS, and more specifically characterizes the message. The request uses the MSG_GET or the MSG_SET type, and the response uses the MSG_ACK type.

OP_STATE_RM_PM - (0x2b)

This OP_STATE_RM_PM operation is used by the SM to transfer performance monitoring state information to or from a RM. The RM responds by acknowledging the receipt and storage of the information segment contained in the message or transfers the current values of its performance monitoring state to the SM. This performance monitoring information will be contained in the data portion of the message. Multiple OPSTATE_RM_PM messages will be sent until the transfer is complete. The value 0x2b is placed in the message operation field. The message subtype is S_RM_STATE, and more specifically characterizes the message. The request uses the MSG_SET type or the MSG_GET type, and the response uses the MSG_ACK type. If the information is contained in multiple messages, the Segment number field is also used.

Message Correlation Number

The message correlation number filed is used to uniquely identify a message. It is used for all (MSG_GET, MSG_SET, MSG_A_INFO, MSG_U_INFO, MSG_A_ACTION, MSG_ACK) message types. It allows the originating entity to associate a response with the appropriate request message. This is obviously not required for a MSG_U_INFO message, but is used for consistency. The responding entity will use the correlation number that it received as the correlation number in the response. This field is required, as multiple messages of the same type/subtype may be outstanding at any time. As an example, if there are three MSG_A_INFO messages outstanding with correlation numbers 100, 101, and 102; three responses are expected with correlation numbers 100, 101, and 102.

Originating Process ID

This originating process ID is a number which uniquely identifies the software process which generated the message. This field is always filled by the message originating process. The responding process uses the value supplied in the request message. This is not a process ID such as that assigned by UNIX or other OS. The value 0x00 is not valid for this field.

Reply Process ID

The reply process ID is a number which uniquely identifies the software process which generated the message response, i.e. the MSG_ACK. This field is filled by the

-48-

process which generates the reply message. In the original message, this field is set to the value 0x00 if the sender does not know the id of the reply process. This is not a process ID such as that assigned by UNIX or other OS. The value 0x00 is not valid for this field.

5 Destination Logical Node Number

The Logical Node Number is the part of the destination address of a message. It specifies the id of a group of shelves which are controlled by a master SM. It will contain a non-zero value for inter-nodal communications. This field must be set to 0x00 for intrashelf messages.

10 Destination Physical Shelf Number

The Physical Shelf Number is the part of the destination address of a message. It specifies the actual physical shelf in a logical node for the operation or information of the message. This field is always non-zero.

Destination Physical Slot Number

15 The Physical Slot Number is the part of the destination address of a message. It specifies the actual physical slot in a shelf for the operation or information of the message. This field is always non-zero.

Destination Physical Port Number

20 The Physical Port Number is the part of the destination address of a message. It specifies the actual physical port in a RM for the operation or information in the message. If this field is unnecessary for a message, the value 0x00 must be used.

Destination Logical Channel Number

25 The Logical Number is the part of the destination address of a message. It specifies the logical channel id in a RM port for the operation or information in the message. If this field is unnecessary for a message, the value 0x00 must be used.

Destination Logical Timeslot Number

The Logical Timeslot Number is the part of the destination address of a message. It specifies the logical timeslot id in a RM for the operation or information in the message. If this field is unnecessary for a message, the value 0x00 must be used.

30 Originating Logical Node Number

-49-

The Logical Node Number is the part of the originating address of a message. It specifies the id of a group of shelves which are controlled by a master SM. It will contain a non-zero value for inter-nodal communications. This field must be set to 0x00 for intrashelf messages.

5 Originating Physical Shelf Number

The Physical Shelf Number is the part of the originating address of a message. It specifies the actual physical shelf in a logical node requesting an operation or information in the message. This field is always non-zero.

Originating Physical Slot Number

- 10 The Physical Slot Number is the part of the origination address of a message. It specifies the actual physical slot in a shelf requesting the operation or information in the message. This field is always non-zero.

Originating Physical Port Number

- 15 The Physical Port Number is the part of the originating address of a message. It specifies the actual physical port in a RM requesting the operation or information in the message. If this field is unnecessary for a message, the value 0x00 must be used.

Originating Logical Channel Number

- 20 The Logical Number is the part of the originating address of a message. It specifies the logical channel id in a RM port requesting the operation or information in the message. If this field is unnecessary for a message, the value 0x00 must be used.

Originating Logical Timeslot Number

- 25 The Logical Timeslot Number is the part of the destination address of a message. It specifies the logical timeslot id in a RM requesting the operation or information in the message. If this field is unnecessary for a message, the value 0x00 must be used.

Message Segment Number

The Message Segment number is used to indicate the sequence of messages for information transfers that require more than the maximum number of data bytes available in one message. As an example, a software download to a RM will probably

-50-

need multiple messages to complete. The message segment number is used to ensure that the data is received in the proper order and without the loss of segments. This field is set to 0x00 for single segment messages. For multi-segment messages, the first segment is numbered 0x01. If the message recipient detects a missing or out of sequence segment, it can refuse any subsequent message segment until the missing segment is received. If missing segment(s) are not forthcoming, the receiving entity should discard any received segments and not perform any part of the requested operation. The sending entity can begin retransmission beginning with the missing segment, restart from the beginning, or cancel the entire message.

10 Status

OK - (0x00)

This status is returned in an MSG_ACK to indicate that the requested operation has successfully completed, or that the information has been received and is valid. The value 0x00 is placed in the field. Non-MSG_ACK messages should always set this

15 field to OK.

FAILED - (0x01)

This status is returned in an MSG_ACK to indicate that the requested operation has failed. The value 0x01 is placed in the field. Further information regarding the failure may be contained in the data field, depending upon the message subtype.

20 READ_ONLY - (0x02)

This status is returned in an MSG_ACK to indicate that the requested operation has failed. The value 0x02 is placed in the field. This status is typically returned to indicate that a MSG_SET message has attempted to modify a variable that is read only. Further information regarding the failure may be contained in the data field, depending 25 upon the message subtype.

BAD_VALUE - (0x03)

This status is returned in an MSG_ACK to indicate that the requested operation has failed. The value 0x03 is placed in the field. This status is returned to indicate that a MSG_SET message has attempted to assign an illegal value to a variable. Further

-51-

information regarding the failure may be contained in the data field, depending upon the message subtype.

NONEEXISTENT - (0x04)

This status is returned in an MSG_ACK to indicate that the requested operation
5 has failed. The value 0x04 is placed in the field. This status is returned to indicate that the variable requested or specified does not exist. Further information regarding the failure may be contained in the data field, depending upon the message subtype.

UNKNOWN_MSG - (0x05)

This status is returned in an MSG_ACK to indicate that the requested operation
10 has failed. The value 0x05 is placed in the field. This status indicates that the message as received is illogical and is unable to be processed. Further information regarding the failure may be contained in the data field, depending upon the message subtype.

BAD_LENGTH - (0x06)

This status is returned in an MSG_ACK to indicate that the requested operation
15 has failed. The value 0x06 is placed in the field. This status indicates that byte count of the message received is not valid. Further information regarding the failure may be contained in the data field, depending upon the message subtype.

IN_PROGRESS - (0x07)

This status is returned in an MSG_ACK to indicate that the requested operation
20 has not yet completed. The value 0x07 is placed in the field. This status indicates that the message received is valid. The original message creator will either have to poll the destination entity at a later time for completion information, or the destination entity that returned the IN_PROGRESS status will generate an autonomous message. These actions are specific to the message subtype.

25 BAD_SUBTYPE - (0x08)

This status is returned in an MSG_ACK to indicate that the requested operation has failed. The value 0x08 is placed in the field. This status is returned to indicate that the message subtype is unknown to the receiving entity. All other MSG_ACK message information will be as received in the MSG_GET, MSG_SET, MSG_A_ACTION, or

-52-

MSG_A_INFO. Illogical subtypes in MSG_U_INFO messages will cause the message to be discarded. Any other action taken by the SM or RM is at their discretion.

BAD_SEGMENT - (0x09)

This status is returned in an MSG_ACK to indicate that the requested operation has failed. The value 0x09 is placed in the field. This status is returned to indicate that the message segment number is illogical, missing or out of sequence in the opinion of the receiving entity. All other MSG_ACK message information will be as received in the MSG_GET, MSG_SET, or MSG_A_INFO. Illogical segment numbers in MSG_U_INFO messages will cause the message to be discarded. Any other action taken by the SM or RM is at their discretion.

Message Byte Count

This field is set to the total length of the message, including both the header and data portions. It must not exceed the value five hundred and twelve (512). If there is no data portion, the value should be set to sixty-four (64) to indicate the length of a header.

15 Message Data

A type/subtype specific data area follows the header. This area can be up to four hundred and forty-eight (448) bytes in length. Some type/subtypes do not currently make use of this area.

CROSS-CONNECTION PROCESSING

20 A cross-connect establishes logical termination point to termination point converts using one or more DS0's/Time slots. The SM 21 (Fig. 6) manages the establishment and teardown of cross-connections. These connections are assigned by the operator through the Operator Interface. The assignment of a cross-connect establishes a data path between two termination points on a system shelf, (e.g. OCUDP 25 to a DS1) termination of a cross connect.

When a cross-connect establishment is initiated, the SM needs

- a) to validate the cross-connection entries (Equipment and Termination points on both ends must be configured and

-53-

- b) allocate TDM bus timeslots from the available pool as shown in Table 2 below.

Table 2. Cross-connect parameters

	CrossConnect Type	# of TDM timeslots allocated	Default signaling	Default Insertion
5	BRI-DSX1 (2B+D) 1B+D 0B+D	3 Tx, 3 Rx* 2 Tx, 2Rx 1 Tx, 1Rx	None (TRSP)	None (TRSP)
	POTS-DSX1	1 Tx, 1 RX (Data) 1 Tx, 1 Rx (Sig)	4-state (D4)	TRB (0x7F)
	OCU-DSX1	1 Tx, 1 Rx**	None (TRSP)	Mux out of Sync
10	SDSL-DSX1	(1 Tx, 1 Rx) times N***	None (TRSP)	None(TRSP)

* BRI termination point usurps up to 3 DS0s depending on BRI service type.

** A second Tx.Rx pair is required if the OCU is configured to operate at 56kbps or 64kbps and running Error correction. Need to be adjacent DS0s with data followed by EC timeslot.

*** N is equal to the configured rate divided by 64k. Corresponds to the number of unchannelized DS0 timeslots needed to transport the SDSL payload. These DS0 Timeslots need to be contiguous.

- In addition, the transmit and receive timeslots assignments of the backplane fabric need to be sent to the resource modules where the termination points are located. The message channel is used to transfer this information. As part of the cross-connect setup, additional parameters need to be defined that relate to trunk processing as shown in Table 3.

Table 3. Cross-connect Parameter Description

	FROM	Access ID (AID) of one termination point that comprises the cross-connect. The FROM side is typically the DS1 side of a DS0 cross-connect. An example of a AID on the DS1 side is: [ShelfID-]CardSlot-Interface-Timeslot
5	TO	Access ID of the termination point that comprises the other end of the cross-connect. The TO side is typically the DS0 side of a DS0 cross-connect. An example of a AID on the DS0 side is: [ShelfID-]CardSlot-Interface
	NSIG	Number of in channel signaling bits that are required to support this connection. A non-zero value indicates the need to assign a pair of TDM timeslots to transport the signaling information across the connection.
10	IW	The insertion word is the 8-bit code that will be transmitted from the Tx timeslot on the DS1 towards the cross-connected DS0 timeslot(s) in the event that the DS1 enters a CGA condition. This can be due to a locally detected Carrier Failure Alarm (LOS or OOF or 10 consecutive SES-Ps) or a remote Carrier Failure Alarm (Incoming Yellow).
10	TC (FROM)	The trunk conditioning parameter determines the value that the in-band signaling bits (AB or ABCD) must be transmitted by the termination point identified in the FROM parameter toward the other end of the connection in the event of a CGA condition as defined previously. This parameter is applicable only for cross-connects requiring in-band signaling. The value of the signaling bits shall be transmitted across the signaling timeslot assigned to this crossconnect.
15	TC (TO)	The trunk conditioning parameter determines the value that the in-band signaling bits (AB or ABCD) must be transmitted by the termination point identified in the TO parameter toward the other end of the connection in the event of a CGA condition as defined previously. This parameter is applicable only for cross-connects requiring in-band signaling. The value of the signaling bits shall be transmitted across the signaling timeslot assigned to this crossconnect.
	PST	This is the primary service state. Setting this value to In Service shall result in the exchange of information across the cross-connect. Setting this to Out of Service shall force IDLE Code to be transmitted from the DS0 toward the DS1. The DS1 shall transmit the assigned or default Insertion Word toward the DS0.

20 The following paragraphs delineate the configurable parameters required to fully define a valid crossconnect.

NUMBER OF SIGNALING BITS (NSIG)

NSIG is a parameter used to define the signaling conventions to be used by termination points on a resource modules at each end of a transmission path. The 25 number of bits specified determines the number of unique states required to support a given signaling convention. The valid settings for NSIG is shown in Table 4. The default setting is based upon the resource module termination point cross-connected. NSIG assumes default values consistent with the type of cross-connect (see Table 2).

-55-

When the resource module is an 4BRI, xOCUDP, or SDSL, NSIG will default to 0 (TRSP - Transparent). Table 4 lists the available NSIG options and the signaling conventions supported.

Table 4. Number of Signaling Bits

5	Signaling Type Name	Description	Signal bit Format	# of signal states
10	TRSP	Transparent. No signaling used.	-	0
	RBS	Robbed-bit signaling.	A	2
	D4	D4 signaling. Used for POTS.	AB	4
	TR008	SLC-96 signaling	AB*	8
	TR303	TR-303	ABCD	16

* TR008 supports a third signaling bit state by allowing the alternation of the B-bit.

IW - INSERTION WORD

The Insertion Word (IW) specifies the 8-bit word directed toward the DS0 inserted upon the detection of a carrier failure. This is normally the termination point's idle code. Table 5 lists and describes the available options.

Not entering a value into IW implies that the operator wants the cross-connect to use the default value as shown in Table 6.

A user-defined hex code pattern can also be specified.
20 IW is valid only when the AID (access ID) for the TO or FROM direction represents a synchronous DS1 resource module.

Table 6 summarizes the contexts in which IW is applicable.

Table 5. Insertion Word

MUX	0x1A.	Multiplex out-of-sync code.
TRB	0x7F.	Standard "trouble" 8-bit idle code.
TRSP	None	Use no insertion word.
Numeric input	Hex value	User-defined insertion word

-56-

Table 6. Insertion Word Defaults Applicability

5

Resource module	Mnemonic	Hex Value
OCUDP, SDSL	MUX	0x1A
4BRI	TRSP	N/A
All other units	TRB	0x7F

TRUNK CONDITIONING

- The trunk conditioning parameter specifies the signaling pattern to be transmitted on DS0 channels being connected, disconnected, or entering a failure condition. It is generated by the either end of a cross-connection toward the other end 10 upon entering a carrier group alarm (CGA) condition. The alarm is encoded as a repeated bit pattern using the signaling bits. Two different patterns are sent. The first (initial) pattern of four bits is sent during the first 2.5 seconds of the CGA condition. The second (final) pattern is sent for the duration of the alarm condition.

- The default trunk conditioning (TC) pattern is a function of the type of 15 termination points comprising the cross-connect. Only cross-connects with non-zero NSIGs perform this trunk conditioning. See Table 7 below for applicable default values.

Table 7. RPOTS Trunk Conditioning Signaling Bit Defaults

20

	NSIG=2 (D4)		NSIG=3 (TR-008)		NSIG=4 (TR-303)	
TC(FROM)	$ABAB_{initial}$	$ABAB_{final}$	$ABAB_{initial}$	$ABAB_{final}$	$ABCD_{initial}$	$ABCD_{final}$
TC(TO)	0 1 0 1	0 1 0 1	0 1 0 1	0 1 0 1	0 1 0 1	0 1 0 1

initial bit pattern transmitted for the initial 2.5 seconds of CGA condition,

final bit pattern transmitted after the initial 2.5 seconds of CGA condition.

-57-

Figure 14 illustrates a POTS DS0 Cross-connect using the following parameters:

FROM AID:	21-2-4
TO AID:	3-2AID: 3-2
NSIG:	2
5 IW:	TRB
TC(FROM):	0101 0101
TC(TO):	NONE

Figure 15 illustrates the effect of Trunk Processing on the cross-connect.

The following describes the signaling channel format for sending information
10 between termination points. The default internal system format corresponds to the D4
signaling format per AT&T PUB 43801 (Ref. 16). In an embodiment of the invention,
two individual timeslots (1 for Rx and 1 for Tx), in every frame, are dedicated to the
signaling associated with an individual cross-connect.

SM Management of Signaling Channel

15 Signaling is used to indicate off-hook/on-hook, ringing, etc., on voice lines. On
DS1 signals, these signaling indications are often embedded in-band via robbed bit
signaling (RBS). RBS ‘robs’ every sixth frame’s LSB for a given voice DSO. This is
fine as long as a reference to the superframe is available (as it is on-board a 2DSX1
RM 110 (Fig. 11). When these in-band robbed bits are carried away from the DSX card
20 via a single voice (PCM) DSO, however, the signaling frame reference is lost, so it is
impossible to determine where the sixth frame is. The LAS 14 implements a stand-
alone signaling channel, with its own framing reference, that is used to carry up to either
a T1 or E1’s worth of signaling bits in real time multiplexed into a single DSO. This
signaling channel also provides a means to transport the raw framing bits from a DS1.

25 The received Signaling Channel is monitored for signaling bits associated with a
particular timeslot. The phase bits (P_0 and P_1) are used to determine both a valid
signaling channel as well as the signaling frame reference. Once a valid signaling DSO
has been verified and the signaling frame reference has been established, signaling bits
may be captured. A valid signaling channel is verified by detecting the repeating Phase

-58-

bit pattern of 00, 00, 00, 00, 00, 00, 01, 01, 01, 01, 01, 01, 10, 10, 10, 10, 10, 10, 11, 11, 11, 11.

At the time a cross-connect is established the SM indicates to the termination points which system timeslots will carry the Tx and Rx signaling information. The SM

5 tracks the following:

- equipment in slot x, termination point y, channel z, rx signaling on system serial data line i, timeslot k
- equipment in slot x, termination point y, channel z, tx signaling on system serial data line j, timeslot m

10 where:

- x designates the slot that contains the resource module
- y identifies the termination point on the resource module (e.g. which DSX-1 on the module)
- z identifies the channel on the interface (e.g. for POTS the termination point and channel are the same)
- i, j identify the serial data line (e.g. SD0 - SD15 of the backplane)
- k, m identify the timeslot that corresponds to the channel signaling information (e.g. timeslot 1-128)

This relationship is shown in Fig. 16.

20 The SM is responsible for administering the association of termination points, cross-connects and timeslots that carry signaling information but does not manipulate the signaling information. This simplifies the tasks which the SM is responsible for, at the expense of system bandwidth.

Signaling Channel Format

25 As noted above, two timeslots are designated to carry signaling, 1 for the transmit direction and 1 for the receive direction. Each timeslot has the format shown in Table 8:

b ₁	b ₂	b ₃	b ₄	b ₅	b ₆	b ₇	b ₈
P ₁	P ₀	S ₁	S ₂	S ₁	S ₄	F	R

30

Table 8. Signaling Channel Byte Format

-59-

where:

- $P_1 P_0$ indicate the phase of the signaling bits
- $S_1 - S_4$ indicate the values of ABCD signaling bits
- F provides the frame alignment signal
- R is reserved

5

In an embodiment S_1 is used to carry the ABCD signaling information of the termination point. $S_2 - S_4$ need not be used. However, they should be used whenever termination points between modules can be grouped together. This will improve the overall efficiency in how the system uses the available bandwidth.

10 Bandwidth Use by System Configuration

The anticipated worst case use of bandwidth for the system is in a 1/0 digital cross-connect configuration. In this scenario the system is populated with DSX-1 modules (Fig. 11). Each module 110 supports two DSX-1 interfaces, each with 24 channels. The system capacity is:

15 $25 \text{ slots} \times \frac{2 \text{ DS1 termination points}}{\text{slot}} \times \frac{24 \text{ channels}}{\text{DS1 termination point}} \times \frac{2 \text{ system timeslots}}{\text{channel}} = 1200 \text{ timeslots}$

To support 1200 timeslots for channel cross-connects, requires an additional 1200 timeslots for signaling.

20 The system capacity is 16 serial data lines $\times \frac{128 \text{ timeslots}}{\text{serial data line}} = 2048 \text{ timeslots}$, a shortage of 352 timeslots.

Additional Signaling Channel Implementation

In order to support the worst case scenario, a more efficient use of the signaling channel format is used for cross-connects to a DS1 termination point. This method is required to convey signaling information downstream from a DS1. The DS0 termination point expects to extract signaling bits using the framing illustrated in Table 9. This requires an extended use of the signaling channel format for the received (i.e., line to drop side, or in this case DS1 to DS0 side) signaling in the following manner:

-60-

		Signaling			DS1 Format	
		2 - State	4 - State	16 - State	SF	ESF
$P_1 P_0$		$S_1 S_2 S_3 S_4$	$S_1 S_2 S_3 S_4$	$S_1 S_2 S_3 S_4$	F	F
	00	A ₁ A ₂ A ₃ A ₄	A ₁ A ₂ A ₃ A ₄	A ₁ A ₂ A ₃ A ₄	F ₁	M ₁
5	00	A ₅ A ₆ A ₇ A ₈	A ₅ A ₆ A ₇ A ₈	A ₅ A ₆ A ₇ A ₈	S ₁	C ₁
	00	A ₉ A ₁₀ A ₁₁ A ₁₂	A ₉ A ₁₀ A ₁₁ A ₁₂	A ₉ A ₁₀ A ₁₁ A ₁₂	F ₂	M ₂
	00	A ₁₃ A ₁₄ A ₁₆ A ₁₆	A ₁₃ A ₁₄ A ₁₆ A ₁₆	A ₁₃ A ₁₄ A ₁₆ A ₁₆	S ₂	F ₁
	00	A ₁₇ A ₁₈ A ₁₉ A ₂₀	A ₁₇ A ₁₈ A ₁₉ A ₂₀	A ₁₇ A ₁₈ A ₁₉ A ₂₀	F ₃	M ₁
	00	A ₂₁ A ₂₂ A ₂₃ A ₂₄	A ₂₁ A ₂₂ A ₂₃ A ₂₄	A ₂₁ A ₂₂ A ₂₃ A ₂₄	S ₃	C ₂
10	01	A ₁ A ₂ A ₃ A ₄	B ₁ B ₂ B ₃ B ₄	B ₁ B ₂ B ₃ B ₄	F ₄	M ₄
	01	A ₅ A ₆ A ₇ A ₈	B ₅ B ₆ B ₇ B ₈	B ₅ B ₆ B ₇ B ₈	S ₄	F ₂
	01	A ₉ A ₁₀ A ₁₁ A ₁₂	B ₉ B ₁₀ B ₁₁ B ₁₂	B ₉ B ₁₀ B ₁₁ B ₁₂	F ₅	M ₅
	01	A ₁₃ A ₁₄ A ₁₆ A ₁₆	B ₁₃ B ₁₄ B ₁₆ B ₁₆	B ₁₃ B ₁₄ B ₁₆ B ₁₆	S ₅	C ₃
	01	A ₁₇ A ₁₈ A ₁₉ A ₂₀	B ₁₇ B ₁₈ B ₁₉ B ₂₀	B ₁₇ B ₁₈ B ₁₉ B ₂₀	F ₆	M ₆
	01	A ₂₁ A ₂₂ A ₂₃ A ₂₄	B ₂₁ B ₂₂ B ₂₃ B ₂₄	B ₂₁ B ₂₂ B ₂₃ B ₂₄	S ₆	F ₃
15	10	A ₁ A ₂ A ₃ A ₄	A ₁ A ₂ A ₃ A ₄	C ₁ C ₂ C ₃ C ₄	F ₁	M ₇
	10	A ₅ A ₆ A ₇ A ₈	A ₅ A ₆ A ₇ A ₈	C ₅ C ₆ C ₇ C ₈	S ₁	C ₄
	10	A ₉ A ₁₀ A ₁₁ A ₁₂	A ₉ A ₁₀ A ₁₁ A ₁₂	C ₉ C ₁₀ C ₁₁ C ₁₂	F ₂	M ₈
	10	A ₁₃ A ₁₄ A ₁₆ A ₁₆	A ₁₃ A ₁₄ A ₁₆ A ₁₆	C ₁₃ C ₁₄ C ₁₆ C ₁₆	S ₂	F ₄
	10	A ₁₇ A ₁₈ A ₁₉ A ₂₀	A ₁₇ A ₁₈ A ₁₉ A ₂₀	C ₁₇ C ₁₈ C ₁₉ C ₂₀	F ₃	M ₉
	10	A ₂₁ A ₂₂ A ₂₃ A ₂₄	A ₂₁ A ₂₂ A ₂₃ A ₂₄	C ₂₁ C ₂₂ C ₂₃ C ₂₄	S ₃	C ₅
20	11	A ₁ A ₂ A ₃ A ₄	B ₁ B ₂ B ₃ B ₄	D ₁ D ₂ D ₃ D ₄	F ₄	M ₁₀
	11	A ₅ A ₆ A ₇ A ₈	B ₅ B ₆ B ₇ B ₈	D ₅ D ₆ D ₇ D ₈	S ₄	F ₅
	11	A ₉ A ₁₀ A ₁₁ A ₁₂	B ₉ B ₁₀ B ₁₁ B ₁₂	D ₉ D ₁₀ D ₁₁ D ₁₂	F ₅	M ₁₁
	11	A ₁₃ A ₁₄ A ₁₆ A ₁₆	B ₁₃ B ₁₄ B ₁₆ B ₁₆	D ₁₃ D ₁₄ D ₁₆ D ₁₆	S ₅	C ₆
	11	A ₁₇ A ₁₈ A ₁₉ A ₂₀	B ₁₇ B ₁₈ B ₁₉ B ₂₀	D ₁₇ D ₁₈ D ₁₉ D ₂₀	F ₆	M ₁₂
	11	A ₂₁ A ₂₂ A ₂₃ A ₂₄	B ₂₁ B ₂₂ B ₂₃ B ₂₄	D ₂₁ D ₂₂ D ₂₃ D ₂₄	S ₆	F ₆

Table 9. Extended Signaling Channel Format

-61-

Notes:

	SF and ESF Format	$F_1 - F_6$ = Frame Alignment Bits
	SF Format	$S_1 - S_6$ = Signaling Framing Bits
	ESF Format	$M_1 - M_{12}$ = Data Link Bits
5	$A_1 - A_{24}$	A bits for channels 1 - 24
	$B_1 - B_{24}$	B bits for channels 1 - 24
	$C_1 - C_{24}$	C bits for channels 1 - 24
	$D_1 - D_{24}$	D bits for channels 1 - 24

- The actual index (1-24) directly corresponds to the number of the cross-connected DS0 timeslot allocated to transport the voice/data information. This reduces the number of timeslots required to support the worst case scenario by 575 timeslots, which then allows the full system fill.

- The signaling channel supports up to 24 separate signaling bit streams, (S_1 through S_4 repeated 6 times per phase, yielding A_1-A_{24} , B_1-B_{24} , C_1-C_{24} and D_1-D_{24}). Making use of the 'Reserved' bit ('R'), adds support for up to 30 signaling bit streams. Signaling bit reference numbers from 1 to 24 remain the same, and the additional 6 signaling bits occupy consecutive R bit positions for $A_{25}-A_{30}$, $B_{25}-B_{30}$, $C_{25}-C_{30}$ and $D_{25}-D_{30}$.

- Upstream Signaling Mapping**
- The following provides the sequence used for inserting signaling information upstream to a DS-1 via the TSI TDM bus 42 (Fig. 6). For example, suppose timeslot 4 on SD1 has been assigned as this DS0's Transmit Signaling slot. Referring to Table 10, notice that in the upstream direction the A/B_{Tx} is always written in the first frame bit position:

-62-

Context	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8
Frame #	P_1	P_0	S_1	S_2	S_3	S_4	R_1	R
5	0	0	A ₁	X	X	X	R ₁	R
6	0	0	X	X	X	X	R ₁	R
7	0	0	X	X	X	X	R ₁	R
8	0	0	X	X	X	X	R ₁	R
9	0	0	X	X	X	X	R ₁	R
10	0	0	X	X	X	X	R ₁	R
11	0	0	X	X	X	X	R ₁	R
12	0	0	X	X	X	X	R ₁	R
13	1	0	A ₁	X	X	X	R ₁	R
14	1	0	X	X	X	X	R ₁	R
15	1	0	X	X	X	X	R ₁	R
16	1	0	X	X	X	X	R ₁	R
17	1	0	X	X	X	X	R ₁	R
18	1	0	X	X	X	X	R ₁	R
19	1	1	B ₁	X	X	X	R ₁	R
20	1	1	X	X	X	X	R ₁	R
21	1	1	X	X	X	X	R ₁	R
22	1	1	X	X	X	X	R ₁	R
23	1	1	X	X	X	X	R ₁	R
24	1	1	X	X	X	X	R ₁	R

 R_1 = INC Reserved for future application

R = Reserved by Bellcore

Table 10.

-63-

An objective is to implement a signaling channel format which maximizes the efficiency of the bandwidth used for signaling. All resource modules can be configured to allow multiple termination point signaling into a single timeslot.

- An alternate approach provides this multiplexing function, gathering signaling from multiple termination points and formatting as depicted in Table 9. As presently described the mapping of signaling from individual termination points to the required network format can be implemented internally by a resource module. However, the multiplexed signaling channel is also made available at the backplane for external examination and verification.
- 5

10 Interpretation of Signaling

The interpretation of the signaling information as conveyed by the signaling channel is as defined in AT&T PUB 62310 and shown in Table 11.

FXS	Loop Start Mode					
	VF input to FXS	Transmit@	Receive@	FXS VF Output		
	A	B	A	B		
Loop Open	0	1				
Loop Closure	1	1				
		1#	*	1	No Ringing	
Loop Open#	0#	1#	*	0	Ringing	
Loop Closure#	1#	1#	*	0	No Ringing	

15

NOTES: @ - Signaling Channel States in the DS1 Signal

* - Designates either 1 or 0

- Indicates Information is qualifier

Blank - Indicates no effect

Table 11.

-64-

Three valid states are defined for any received signaling bit '0', '1', and 'Alternating'.

The RM logic detects any one of these three states. For each bit, (A, B, etc.), the following guidelines are used:

- After a minimum of four consecutive 1's, a '1' signaling bit is declared.
- After a minimum of four consecutive 0's, a '0' signaling bit is declared.
- After a minimum of four consecutive alternating bits (i.e., 0101, or 1010), an 'Alternating' signaling bit is declared.
- All other patterns of signaling bits are declared 'Indeterminate'.
5

Up to four signaling bits can be monitored A, AB, or ABCD. Each bit may be
10 detected as either a '0', a '1', 'Alternating', or 'Indeterminate'. Note that using the above rules, a valid detection can be made within 3 msec. If an input signaling stream is declared 'Invalid', all receive signaling bits associated with that stream indicate 'Indeterminate'.

Loop Start Scenario:

15 RPOTS detects On-Hook (Loop Open) condition Asserts Transmit Signaling Bits A/B_{Tx} = 0/1
RPOTS detects Off-Hook (Loop Closure) condition Asserts A/B_{Tx} = 1/1

RPOTS detects Receive Signaling Bits A/B_{Rx} = */1 No ringing to loop regardless of A/B_{Tx} state
RPOTS detects Receive Signaling Bits A/B_{Rx} = */0 No ringing to loop if A/B_{Tx} = 1/1 (Off-hook)
RPOTS detects Receive Signaling Bits A/B_{Rx} = */0 Send ringing down loop if A/B_{Tx} = 0/1 (On-hook)

20 where A/B_{Tx} are transmitted from RPOTS across the assigned Tx timeslot.
A/B_{Rx} are received by RPOTS across the assigned Rx timeslot.

Graphical User Interface

The following describes a Graphical User Interface (GUI) Management System for the local access system (LAS).

25 The Management System (MS) of the present invention performs a variety of functions for the user. These functions include: Configuration Management, Equipment

-65-

- and Facility Testing, Cross-Connect Mapping, Status Tracking, Event and Alarm Reporting, and Statistics Reporting. These functions are expected to be easy to use and provide graphical at-a-glance information. This is accomplished through the use of color, shapes and sound to convey and receive information. The user does not need to
- 5 perform complex operations in order to effectively operate the MS or to manage equipment in the LAS. The MS is graphical and context sensitive, making it easy to use and easy to understand.

The MS provides a hierarchical methodology by which the user can operate, maintain, and monitor the LAS. Consistent presentation format and context sensitive

10 operations simplify the management tasks. The rigorous maintenance of open windows and presentations provide the most clear and concise information format for the user. As an example, all user-initiated changes are marked in red for enhanced visibility before the user commits the changes to the system. Operations are consistent in their use, and provide little opportunity for user error.

15 The following graphical user interface (GUI) terminology is used herein:

Window	Window refers to a rectangular box on the computer display
Pop-up Window	A pop-up window is a window that suddenly appears as a response to a mouse or button click.
Dialog Box	A dialog box is a pop-up window that permits additional information to be supplied by the user.
Message Box	A message box is a pop-up window that displays important information from the MS. Message boxes usually display warnings or error messages.
20 Check Box	A check box allows the user to toggle a choice.
Button	Clicking on a button executes the command that is written on the button.
Radio Button	Radio buttons allow the user to select an option out of many options.

-66-

Field	An area in a window or dialog box where either the user can enter information or the system displays information.
Drop-down List	A list of items that appears when the user clicks on the down-pointing arrow located to the right of the drop-down list.

How the System is Displayed and Organized

- 5 The MS is displayed and organized in a hierarchical structure which comprises a 3x3 matrix of columns and rows. The function views--operational view, cross-connect map view, and administration view--form the column headers of the MS matrix. Each of these views supports the following entity levels, as row headers: shelf, equipment, and interface (termination points). This modular approach makes the system easy to
- 10 understand. Each display is also well-organized, clear, and easy to view. As a result, it is relatively simple for a user to navigate through the system by clicking on any of the function views at any time, and then drilling up or down to different entity levels. For example, the user can configure the timing source of the shelf manager (SM) by clicking the operational view button and then drilling down to the equipment screen. This
- 15 approach and layout greatly facilitates the operation, maintenance, and monitoring of LAS shelves.

The 3x3 matrix for the MS is shown below:

	Operation	ConMap	Admin
Shelf	Operational View - Shelf	Cross-Connect Map View - Shelf	Administration View - Shelf
Equipment	Operational View - Equipment	Cross-Connect Map View - Equipment	Administration View - Equipment
Interface	Operational View – Interface	Cross-Connect Map View – Interface	Administration View – Interface

Screen Layout

Referring to FIG. 17, the MS display is shown divided into four areas or zones. From top to bottom, the zones are labeled zone A, zone A1, zone B and zone C. The system has been designed so that all zones are always visible and cannot be obscured by 5 other windows. The advantage of this windowing structure is that the view of both specific information and overall system information is always available to the user. A further advantage is that the user's computer screen is never cluttered with small windows and dialog boxes.

Zone A, shown in FIG. 18, contains the following elements: toolbar 202, shelf 10 drop-down list 204 and session indicator 206. Each of these zone A elements is now described.

The toolbar is a panel of icon buttons which represent commands or controls. There are three groups of buttons: navigation 200, view 210, and control 212. Each group provides quick access to commonly used functions as defined in the following 15 Table 12:

Group	Name	Definition
Navigation	Home 201	Go to the shelf level screen. Highest entity level of the NMS.
Navigation	Back 203	Go to the previous screen. This operation is valid only when the user has navigated forward.
Navigation	Forward 205	Go to the next screen. This operation is valid only when the user has navigated forward and has navigated back.
20	Navigation Up 207	Go up one entity level.

-68-

		Clicking this button activates the operational functions. Provisioning, status/alarm reporting and maintenance of the shelf elements are available here and are displayed in Zone B.
		Clicking this button activates the mapping functions for Cross-Connections. The user can provide connections between the various pieces of equipment, interfaces and timeslots.
		Clicking this button activates the system administration functions which include user administration, software management and data backup.
		Clicking this button uploads/downloads information from a LAS shelf. The icon changes to end session upon activation of the session.
		Clicking this button terminates the current working session.
5	Control	Help 217 Clicking this button activates the Online Help system.
	Control	Exit NMS 219 Clicking this button exits the MS.

TABLE 12

Shelf Drop-down List 204

- The shelf drop-down list is a list of shelf node identification numbers. Clicking 10 on the down-pointing arrow 214 allows the user to select the shelf that the user wishes to monitor.

-69-

Session Indicator 206

The session indicator has the following color coded meanings in Table 13:

Color	Definition
5	GREEN A communication link is active with the shelf and the operator is logged into the system.
	YELLOW A communication link is active, but the operator is not logged in.
	RED The SM communication link is currently down.
	GRAY The MS is currently operating in an off-line mode.

TABLE 13

The following table describes transition function for the session indicator:

From	To	Reason	Resultant Operation
RED	YELLOW	Login Sequence Started (Default)	Display Login Window
YELLOW	GREEN	Authentication Successful	Information from the SM has been retrieved and displayed on the PC screen.
GREEN	RED	Session Terminated	OFFLINE Mode
YELLOW	RED	Session Terminated	OFFLINE Mode

15 TABLE 14

Zone A1, shown in FIG. 19, contains shelf descriptor 220 and screen level descriptor 222 information. The shelf descriptor 220 identifies the current shelf, slot and interface for the screen that is displayed. Additional information may appear in zone A1, depending upon the screen that the user is viewing. The screen level descriptor 222 identifies the current screen that is displayed in terms of the particular

-70-

view (operational, cross-connect map or administration) and entity level (shelf, equipment or interface).

Zone B, shown in FIG. 20, includes content which varies depending upon the current function. Most user interaction occurs in zone B. The first screen represents the 5 operational view - shelf. The screen content of this area is determined by the active view button located in zone A. All zone B screens have the following levels of visibility into the system:

	Top Level	Shelf
	2 nd Level	Equipment (RM or SM)
10	3 rd Level	Interface

When the user successfully completes login to the system and clicks on the operational view button, the user sees all modules in the current shelf at the top level. Next, if the user clicks on a particular module, a blow-up view of the selected module as well as provisioning information associated with this module is presented; this is the 2nd Level.

15 Finally, when the user clicks on any interface of the selected module, the user is brought to the 3rd Level; from here, the user can access interface information.

Zone C, shown in FIG. 21, includes equipment and interface LEDs. The following Table 15 describes the LED states:

	Indicator Color	Operational Status
20	RED	There is a standing alarm or fault condition on one or more equipment or termination points in the shelf.
	YELLOW	One of the following conditions has been detected on the module or interface located in that slot: an active maintenance session (loopback) is in effect, or a module is administratively out of service.
	GREEN	There are no standing alarms or fault conditions on equipment located in that slot or the termination points that reside there.

ORANGE	Indicates that the interface or module is autonomously out of service (i.e., placed out of service by the user OR a situation has occurred that implies that this interface or module is out of service).
Light GRAY	Module is not present and is not configured.
Dark GRAY	Indicates one of the following conditions exists at that slot location: module is configured but not present, or module is present but not configured.

TABLE 15

- 5 To log into the MS, the user chooses the shelf ID to configure from the shelf drop-down list and clicks on the start session button. A security window appears which requests the user to input user name and password. After the user enters the user name and password, a database master selection appears (not shown) with options to either extract configuration information from the shelf or to use configuration information stored in a
- 10 MS database. Upon the user making a selection, the system starts data-synchronization such that SM information is uploaded and displayed on the PC screen. Until this process completes, the user is locked out of any operation in the MS. Once the data synchronization process is completed, the operational view – shelf screen appears as previously shown in FIG. 17.

15 Configuration Overview

The MS provides a real-time representation of the LAS system. All of the LAS module types, e.g., SDSL, 2DSX1, 4BRI, 4POTS, 4POTSR, 2CPOTS and 2OCU, are represented and their operating conditions are displayed. Note that before the user commits any changes to the system, all changes are marked in red for enhanced visibility. The MS supports asynchronous connection to one LAS shelf at one time.

-72-

Equipment Level

The equipment level display Zone B, second level, provides the larger, equipment representation. Additionally, a config tab is provided which shows the current equipment configuration, and a means for changing the parameters as necessary.

- 5 The installed equipment type is shown when it differs from the requested configuration. An equipment configuration timestamp and a display of the user who performed the most recent configuration change are also provided. From the equipment display, the user can select a particular interface by clicking on the button (1-4) associated with it. This moves the user to the interface level.

10 Interface Level

The interface level display Zone B, third level, also shows the larger, equipment representation. A Config tab is provided which shows the current interface configuration parameters, and a means for changing the parameters as necessary. An interface configuration timestamp and a display of the user who performed the most recent configuration change are also provided.

Configuring the SM

To configure clock management in the SM, the installed equipment type is auto-detected and shown in the installed eq type field. The SM is pre-configured on the 26th slot. To setup the timing source, the user clicks the SM slot in zone B and then clicks

- 20 the config tab 230 as shown in FIG. 22. A selection is made from options for primary, secondary and tertiary synchronization sources as shown in FIG. 23. Note that the current timing field 232 displays the active timing source. Examples of configurations for some of the modules is now described.

Configuring SDSL

- 25 The SDSL module receives configuration and optioning information from the SM via a control interface over a backplane bus on the LAS shelf. To configure equipment type, the user clicks the slot with the SDSL label in the operational view – shelf screen, clicks the config tab 230 and then clicks SDSL in the eq type drop-down

-73-

list as shown in FIG. 24. The selectable user data rates of the SDSL interface include 256 kbps, 384 kbps, 512 kbps, 768 kbps, and 1152 kbps.

- To configure interface properties for the SDSL module, the user clicks the interface button in zone B or interface LED in zone C, clicks on the config tab 230, 5 selects the data rate in the data rate drop-down list 232 and selects the sealing current 234 ON or OFF option as shown in FIG. 25.

Configuring 4POTS/4POTSR

- The 4POTS module is useful for long loop applications with bulk ringing availability from an external ringing generator. For short loop applications, without 10 bulk ringing availability, the 4POTSR module with on-board ringing capabilities can be used. To configure a 4 POTS equipment type, the user clicks the slot with 4POTS label in the operational view – shelf screen, clicks the config tab and then clicks 4POTS in the eq type drop-down list. To configure interface properties, the user clicks the interface button in zone B or interface LED in zone C, clicks on the config tab 230 and 15 selects the transmission type 236 in the on hook transmission drop-down list as shown in FIG. 26.

Event and Alarm Reporting

- Status changes, alarms and all user initiated changes result in the generation and timestamped logging of events. These events provide an audit trail for the operation of 20 the LAS system. In the case of user initiated changes, the user who performed the change is also noted in the event (see FIG. 27). The MS user can select a presentation format, filter events by type (e.g. major, minor, etc.), time of occurrence, and number of events to display. The user can also select which information is presented and the order in which it is displayed. Additionally, the displays can be sorted in ascending or 25 descending order, or not at all. These options can be permanently saved for future application to the events, or modified on a temporary basis for a specific need (see FIG. 28).

-74-

Shelf (Shelf Manager)

The shelf that the user selects provides the basic context for the display. All events associated with all of the equipment and interfaces in the shelf are displayed.

Equipment Level

- 5 Typical equipment events are generated and logged for equipment insertion and removal, status changes, and configuration changes. The selected equipment provides the basic context for the display. Only events associated with the selected piece of equipment are displayed.

Interface Level

- 10 Typical interface events are generated and logged for status changes, alarms, and configuration changes. The interface provides the basic context for the display. Events associated with the selected interface are displayed.

Cross-Connection Mapping

- 15 The connections between the various pieces of equipment and interfaces and timeslots are managed with this feature. As in the cases above, this information is selected, displayed and modified with the LAS operational hierarchy in mind.

Shelf Display

- The shelf display provides the user with a representation of the connections between the pieces of equipment resident in a shelf. The user selects a slot, and the associated connected pieces of equipment are highlighted (See FIG. 29). If more detailed connection information is desired, the equipment can be double-clicked.

Equipment Display

- The equipment display (see FIG. 30) provides the user with a representation of all of the connections associated with the interfaces of the selected equipment. A live, detailed representation of the equipment is also provided. The user can select particular connections of interest in order to view the equipment at the other end of the

-75-

connection. When the desired connection is found, it can be double-clicked to narrow the focus to the interface.

Interface Display

The interface display (see FIG. 31) provides the user with a representation of all 5 of the timeslots associated with the connection on the selected interface. A live, detailed representation of the equipment is also provided. The user can select particular timeslots of interest in order to view the equipment at the other end of the connection. When desired timeslots are identified, an opportunity is provided for the user to change the connection.

10 Active Operating Condition

The MS provides active operating condition and equipment presence displays. This display accurately depicts the shelf and all of its cards along with their operating condition, including the detection of cards inserted, removed, configured, etc. System card types, including SDSL, DSX, BRI, POTS, POTSR, and OCU are represented.

15 Shelf Display

The shelf display provides a representation of an LAS shelf. This includes all operating LEDs, and card representations etc. Objects are created that resemble the actual hardware, thus making it easy for a user to recognize components and use the system. From the shelf display, the user can select a particular piece of equipment by 20 clicking on it. By clicking on a slot in the shelf view, the user moves to the equipment level which provides a more detailed look at the selected equipment.

Equipment Display

The equipment level display provides a larger, even more detailed equipment representation than the shelf display, including labeling. Additionally, a Status tab is 25 provided which shows the equipment status (In Service, Out of Service), and time of the most recent status change. A method is provided to manually change the equipment status. A display of the user who performed the most recent manual status change is

-76-

also provided. From the equipment display, the user can select a particular interface by clicking on it (Buttons 1-4). This moves the user to the interface level.

Interface Display

The interface level display provides the same detailed equipment representation, 5 including labeling. A Status tab is provided which shows the interface status (In Service, Out of Service) and the time of the most recent status change. A method is provided to manually change the interface status. A display of the user who performed the most recent manual status change is also provided.

Several views from the matrix of views are now described.

10 Operational View - Shelf

The operational view - shelf screen displays a realistic view of the front of the shelf as shown in previous FIG. 17. This screen shows 26 clickable objects, each representing a slot in the active shelf. Each object includes an equipment type text label and LEDs applicable to the installed equipment type. LED display areas whose status is 15 presently not communicated to the SM are grayed out. Shading and LED colors of the slots are based on four possible conditions:empty and unprovisioned, empty and pre-provisioned, RM inserted and unprovisioned, and RM inserted and provisioned.

The control functions pertaining to this screen are as follows. Each equipment faceplate area is an active area that, when clicked, forces a zoom to the Operational 20 View - Equipment screen. The user can also click on the numbered buttons located beneath each faceplate to achieve the same result. A single left click forces a transition to the Operational View – Equipment -Status tab applicable to that equipment in Zone B.

Operational View - Equipment: Resource Module

25 The operational view - equipment: RM screen displays the representation of the equipment faceplate and permits the user to perform the following operations: enter or modify the equipment type; place the equipment in and out of service; review the

-77-

- complete attribute set of the equipment; and transition to the next level, the interface level as shown in FIG. 32. This screen's Shelf Descriptor 220 "Shelf 1(Central Office) Slot 24" and Screen Level Descriptor 222 "Operational View – Equipment" are displayed in Zone A1. In this screen, Zone B consists of two components: RM 5 faceplate 240 and tabbed dialog boxes 242.

- The faceplate is an actual representation of an RM. In this example, its name, 2DSX1, appears at the top of the faceplate. Each LED is a real-time display indicator for this RM. At the bottom of the faceplate is this RM number, e.g., 24. The two I/F buttons 250 associated with this RM are shown directly to the right of the faceplate. 10 The user can drill-down a level to view and change information associated with this RM, by clicking on an I/F button. For example, the user can click on I/F 01 button to the interface level for I/F 01.

- To the right of the faceplate is a larger dialog box organized into four tabs 252: Status, Config, Event, and Perf. The following functionality is contained in each tab. 15 Items that are read-only are indicated by "RO". Read-write fields are indicated by "RW".

- Status
- Condition (RW)
- Options: In Service, Out of Service
- Change Date and Time (RO)
- Previous Condition (RO)
- Installed/Removed (RO)
- Most Recent Manual Status Change Date/Time and User ID (RO)
- Config (Configuration)
- Equipment Type (RW)
- Options: None, 4BRI, 2DSX1, 2OCU, SDSL, 4POTS, 4POTSR, 4POTS+, 2T1, 2T1+, 2SDSL, SRU, EPRM2
- Software Version (RO)
- Number of Channels (RO)
- Installed Equipment Type (RO)
- Most Recent Manual Status Change Date/Time and User ID (RO)

-78-

Event

- Date/Time (RO)
- Slot (RO)
- Interface (RO)
- 5 Severity (RO)
- Event Type (RO)
- Perf (Performance)

Operational View - Equipment: Shelf Manager

- The description of the operational view - equipment: SM screen, Shelf
- 10 Descriptor "Shelf 1(Central Office) Slot 26" and Screen Level Descriptor "Operational View – Equipment", are displayed in the Zone A1. In this screen, Zone B consists of two components: SM faceplate 240 and tabbed dialog boxes 242 as shown in FIG. 33.
- The faceplate is an actual representation of an SM. In this example, its name, SM, appears at the top of the faceplate. Each LED is a real-time display indicator for
- 15 the SM. At the bottom of the faceplate is the SMU's slot number, 26.
- To the right of the Faceplate is a larger dialog box organized into four tabs 252: Status, Config, Event, and Perf. The following functionality is contained in each tab. Items that are read-only are indicated by "RO". Read-write fields are indicated by "RW". Status
- 20 Condition (RW)
- Options: In Service, Out of Service
 - Change Date and Time (RO)
 - Previous Condition (RO)
 - Installed/Removed (RO)
- 25 Most Recent Manual Status Change Date/Time and User ID (RO)
- Config (Configuration)
- Equipment Type (RO)
 - Shelf Name (RW)
 - Current Timing (RO)
- 30 Sync Source (RW)

-79-

- Primary RM
 - Options: 1-25; OFFICE; NONE
- Primary I/F
 - Options: 1-4; NONE
- 5 Secondary RM
 - Options: 1-25; OFFICE; NONE
- Secondary I/F
 - Options: 1-4; NONE
- Tertiary RM
 - Options: 1-25; OFFICE; NONE
- 10 Tertiary RM
 - Options: 1-4; NONE
- Activate (RW)
- ACO (RW)
- 15 Most Recent Manual Status Change Date/Time and User ID (RO)
- Event
 - Date/Time (RO)
 - Slot (RO)
 - Interface (RO)
- 20 Severity (RO)
- Event Type (RO)
- Perf (Performance)

- Operational View - Interface
 - The description of the operational view - interface screen, Shelf Descriptor
- 25 220 "Shelf 1(Central Office) Slot 24 Interface 1" and Screen Level Descriptor 226
"Operational View -Interface", is displayed in Zone A1. In this screen, Zone B consists
of two components: RM faceplate 240 and tabbed dialog boxes 242 as shown in FIG.
34.
- 30 The faceplate is an actual representation of a RM. In this example, its name,
2DXS1, appears at the top of the faceplate. Each LED is a real-time display indicator for

-80-

this RM. At the bottom of the faceplate is this RM's number, 24. The two I/F buttons associated with this RM are shown directly to the right of the faceplate. Note that the color of I/F 01 is black. This indicates that I/F 01 represents the current view. By clicking on the blue I/F 02 buttons the user can view data specific to interface 2. When
5 the user clicks on I/F 02 its color changes to black, indicating that this is the current view.

To the right of the faceplate is a dialog box organized into five tabs 252: Status, Config, Event, Perf, and Maint. The following functionality is contained in each tab. Items that are read-only are indicated by "RO". Read-write fields are indicated by
10 "RW".

- Status
 - Condition (RW)
 - Options: In Service, Out of Service
 - Change Date and Time (RO)
 - 15 Previous Condition (RO)
 - Installed/Removed (RO)
 - Most Recent Manual Status Change Date/Time and User ID (RO)
 - Config (Configuration)
 - Framing Format (RW)
 - Options: ESF, SF
 - Line Code (RW)
 - Options: B8ZS, AMI
 - Equalization (RW)
 - Options: 0 to 133 ft, 133 to 266 ft, 266 to 399 ft, 399 to 533 ft, 533 to
25 655 ft
 - Most Recent Manual Status Change Date/Time and User ID (RO)
 - Event
 - Date/Time (RO)
 - Slot (RO)
 - 30 Interface (RO)
 - Severity (RO)

-81-

- Event Type (RO)
- Perf (Performance)

Cross-Connect Map View - Shelf

- From the Shelf screen, the user clicks on the ConMap Icon 211; then clicks on 5 one of the modules in Zone B to see its connections. When the user does this, the color of this module changes to blue. All other modules that are NOT connected to the "blue" module change color to light gray. The modules that are connected to the "blue" (Connected-From module) module do not change color; they retain their original color. In this example, we see that modules 7 and 24 are connected to each other. To see an 10 exploded view of what's connected to module 7, the user simply clicks on module 7 as shown in FIG. 35.

Cross-Connect Map View - Equipment

This screen is arrived at by clicking on board 7 in the "Cross-Connect Map View - Shelf" screen.

15 Connected-From and Connected-To Interface Slots

- The faceplate for board 7 is displayed in Zone B1 and its interface connections in Zone B-middle Fig. 36. At this time, nothing is displayed in Zone B3. The currently selected interface button is colored black. (The user can select an interface button by clicking on it.) Non-selected buttons are blue. Slot 7 is shown, with its four interface 20 buttons 260, in the center of B-middle. Around them is a button 262 labeled 24-01. There is also a blue line 264 connecting buttons 07-01 and 24-01. This indicates that interface 01 on slot 24 is connected to interface 01 on slot 07. In this case, slot 7 is the Connected-From slot 266 and slot 24 is the Connected-To slot 268 (Fig. 37).

Viewing the Connected-To Faceplate

- When the user clicks on one of the Connected-To interface buttons, the faceplate of the slot associated with this interface button appears in Zone B3; in this case it's slot 24. In addition, the color of the connecting line between the two sets of interface buttons 25

in Zone B2 changes to black, and the color of the Connected-To interface button also changes to black. The purpose of the color change is to highlight the selected connection. By clicking on each Connected-To Interface button, the user can see its associated faceplate as shown in FIG. 36.

5 Cross-Connect Map View - Interface

To drill down from the equipment screen to the next level, the user clicks an enabled interface button alongside a faceplate in Zone B3. This brings the user to the Cross-Connect Map View - Interface screen. Zones B1 and B3 retain their original faceplate images. However, Zone B-middle has changed. It now displays a timeslot matrix 270, indicating the "From TS" and "To TS" timeslot buttons 266, 268. (TS stands for Time Slot.) The color of each button in the matrix is blue, indicating that the user has not selected any button for further viewing. In addition, the number of the Slot, Interface, and Timeslot are printed on each button. For example, the button in column 1, row 2 that is labeled 07-01-02 indicates that this is the timeslot associated with slot 7, interface 1, timeslot 2. These features are shown in FIG. 37.

Selecting TimeSlots

To drill down to a further level, the user clicks on one of the timeslot buttons 270. The user will notice that the color of the "From TS" and "To TS" timeslots that the user has chosen will have changed color to black. Additionally, any other timeslot buttons that are grouped with the connection that the user has chosen, if any, are also black. When this happens, the user will see the Conn/Disc Sel'd TS's button 280 appear. This button caption stands for "Connect/Disconnect Selected TimeSlots" as shown in FIG. 38.

Cross-Connect Map View - Interface: Connection Dialog Box

The user can click on the Conn/Disc Sel'd TS's button to bring up the Connection Dialog Box as shown in FIG. 39. The top portion of this dialog box shows the current starting timeslot 282, ending timeslot 284, and number of affected timeslots 286 information. The number of affected timeslots is 3, in this example, because the timeslots begin with 1 and end with 3. The Disconnect button 290 permits the user to

-83-

disconnect currently connected timeslots. Disconnect is enabled for a valid connection only. The Connect button 288 permits the user to connect timeslots that do not have any current connections.

- If the user clicks the Lock button 292, the system will "remember" timeslots. It
- 5 is enabled only for Interfaces that have no associated connections. This will be obvious when the user sees the word "unassigned" in the "To TS" timeslot area. The purpose of the Lock button is ease of use. After the user clicks the lock button, the user can perform other operations in the MS and then return to this screen; the system "remembers" the locked timeslots. For example, the user can choose other timeslot(s) that the user would
- 10 like to connect to the locked one(s). To do so, however, requires navigating to another screen and returning. When the user returns, the "remembered" timeslot—as well as the newly selected one—will appear. The user does not need to re-enter the information.

Changing Parameters

- A connection exists between two timeslots if the word "Unassigned" does NOT
- 15 appear in any of the "From TS" and "To TS" timeslots. When a connection exists, as in this case, a number of user-changeable input boxes will appear in the bottom of this dialog box as shown in Zone B (middle) of FIG. 40.

Administration View - Shelf

- The administration view -shelf screen displays a realistic view of the front of the
- 20 shelf as shown in FIG. 40/41. There are 26 clickable objects 294, each representing a slot in the active shelf. Each object includes an equipment type text label and LEDs applicable to the installed equipment type. LED display areas whose status is presently not communicated to the SM are grayed out.

Administration View - Equipment

- 25 Clicking the resource module in the Administration View – Shelf screen will display the Administration View – Equipment screen in Zone B. In this screen, RM's faceplate and three-tabbed (User Profiles, Archive, and Restoral) dialog box 296 are displayed as shown in FIG. 42. The user can change an existing user ID or password by

-84-

entering the appropriate information in the user profile dialog box 298. After the user has made all changes in the MS, the user can archive the database to hard-disk by clicking the Backup Database NOW button 295. The user can restore the database by clicking the Restore Database NOW button 297.

5 Administration View - Interface

The administration view - interface screen supports the functions as the Administration View – Equipment screen as shown in FIG. 43.

Software Architecture for the MS

This section describes some of the requirements which affect the architecture chosen for
10 this software. To the user, the MS is a single executable program, with supporting DLLs, database file, and an INI file, which runs on a PC using Windows 95, Windows 98, or Windows NT 4.0. The program provides monitoring of the LAS equipment, as well as the ability to configure and control the equipment, all through a single, consistent user interface.

15 The user interface presents a single, well-defined window through which all user interaction takes place, providing a common framework for all monitoring and configuration operations. The common framework helps the user learn the MS more quickly by repeating a familiar framework during all operations. A PC is attached to the LAS equipment via a cable (serial or LAN). Communications with the LAS hardware
20 occurs transparently to the user, and the results of these communications are automatically reflected in the user interface (i.e., each display is "live", and is updated without user intervention, if the data represented on the components changes).

Context Diagram (Fig. 44)

This section describes the context in which the MS software operates. As noted
25 above, the PC 300 on which MS executes is attached to LAS equipment 14 via a cable 302 (serial or LAN). The user receives output from the MS through a single window presented on the display 308, and interacts with the software primarily by using the mouse 304, with keyboard 306 input at times (e.g. entry of username and password).

-85-

Various items of information are stored permanently on the PC hard drive for the program's use. These items include a configuration database for the LAS equipment, user-selected options and program configuration settings, and configuration information about the communication link(s) between the PC and LAS.

5 Data Flows (Fig. 45)

Due to the requirement that the user interface displays must be "live", i.e., automatically updated in real-time to reflect the condition of the attached equipment, several separate threads of execution are used in the software to isolate the user interface from the subsystems which communicate with the LAS equipment. A top-level data flow diagram FIG. 45 indicates the data that flows between the threads and to or from the external elements. As shown, the *User Interface*, *Comm Reader*, and *Comm Writer*, are each separate threads of execution.

Database Description (Fig. 45)

The *Database* 310 contains configuration, condition, event, and historical data regarding the LAS equipment managed by the MS. This data is made available to the *User Interface* 312 for the user to control, based on the requirements imposed on the user interface. The *Data Manager* 314 module provides an encapsulated interface to the Database, and provides intelligent get/set functions which handle many of the internal consequences of changing each piece of data.

20 The data structures in the database are object-oriented structures, nested in a hierarchy which mimics the hierarchy of LAS equipment. At the top level, there is a *Node* object, which represents a group of *Shelf* structures. Each Shelf is composed of a number of *Slot* structures, and each Slot contains a number of *Port* structures. There are several types of Ports, with all Ports in a Slot presumed to be of the same type.

25 Thus, the specific knowledge of LAS equipment is modularized into just a few classes. This modularization can be taken further, such that each equipment type can be represented throughout the system by a single class. This encapsulation is very maintainable, and very modifiable, reducing the cost and elapsed time involved in making additions in the future.

-86-

For each Shelf, there is an *Event History*, which contains all of the events which have occurred to elements of this Shelf. In addition, each Shelf also has *Timing Info*, which describes how the clock signals for the Shelf are obtained and managed.

- The MS is implemented in Microsoft Visual C++ for execution on Microsoft
- 5 Windows 95 and Windows NT 4.0. The Microsoft Foundation Classes (MFC) application framework is also utilized to leverage the advanced GUI elements already present in Windows and Win32 into the MS software. Visual C++ is a defacto industry standard, since it is the most widely used software development platform in the world. The MFC application framework provided with Visual C++ is likewise well known.
- 10 As with other VC++/MFC applications, porting to UNIX or other operating systems, integration with Java-based applications, etc. is easily achieved without the limitations present with other languages.

- Communication with the LAS equipment is performed in separate threads of execution from the User Interface thread. This keeps the user interface responsive, and
- 15 also allows it to receive updates from the attached equipment at the same time as the user is performing some operation. Encapsulating the communications in this way also modularizes the communications subsystem, so that it can easily be modified or upgraded. This is especially useful if either the physical or data link layers used to communicate with the LAS equipment are changed.

20 Fiber Access to a Resource Module

Certain of the resource modules 10 are required to have a fiber optic interface. In order to preserve the interchangeability of modules the fiber access to these modules must be common to each of these special modules to achieve the goal of providing any type of service in any shelf slot. Therefore, in accordance with a preferred embodiment

25 of the invention as shown in Fig. 46, a fiber optic cable entry slot or window 401 (about 1" in length) is provided in an upper rail 402 of each module chassis 404 to enable a fiber 406 to be inserted from above the shelf (not shown) and between slots in the shelf to mate with a fiber optic transceiver 408 (to the extent provided in the module). The basic method of routing the cable is shown in Fig. 46. The roof of the shelf provides a

method of securing the cable near the entry point. A fixed amount of cable slack is made available to reach the transceiver 408 located near the lower half of the backplane connector 410. The cable slack provided by the loop 412 is sufficient to allow the RM 10 to be partly removed without applying stress to the cable 406. A mechanical stop mechanism 414 prevents the RM from being removed before damaging stress can be applied to the fiber/cable. An additional optional element (not shown) is a shroud, which would prevent the loose cable from being caught on surface components.

5 Particular attention must be given to component heights within the loose cable region. Sufficient slack is present in the cable to guide the cable along the top edge of the

10 Printed Circuit Board (PCB) 416 during removal to avoid tall components on the PCB 416.

The following advantages are achieved in this embodiment:

- Allows the faceplate design and indicator LEDs (not shown) on the faceplate 417 set to be consistent with that of all other RM's;
- 15 • Can be retrofitted in the field;
- Allows large bend radius of fiber 406;
- Allows fiber to be deployed from the rear of the unit;
- Any number of fiber interfaces can be accommodated;
- Other media may be accommodated (i.e. DS3 coax, RJ48, etc.)

20 The mechanical stop mechanism 414 should engage the lower rail of the shelf during removal of the RM 10 and prevents further displacement by gravity forced engagement with a stop 420 on the bottom wall 404 of the special module 10.

One-Wire Protocol and Message Channel

A one-wire serial interface connects each RM slot to the SM slot via the

25 Backplane. Before a module 10 can communicate a message from the Message Channel granting access to the Backplane must be delivered. One interface is dedicated to each RM.

-88-

This section describes the one-wire start-up and take-down protocol interface used to grant backplane access to a newly inserted module, or removal of backplane access from a faulty module. Chart 1 diagrams the steps required to grant backplane access protocol. These steps are implemented using a Dallas Semiconductor Chip

5 described in (Reference 15) Data Sheet DS2405, which has been enhanced in accordance with the invention.

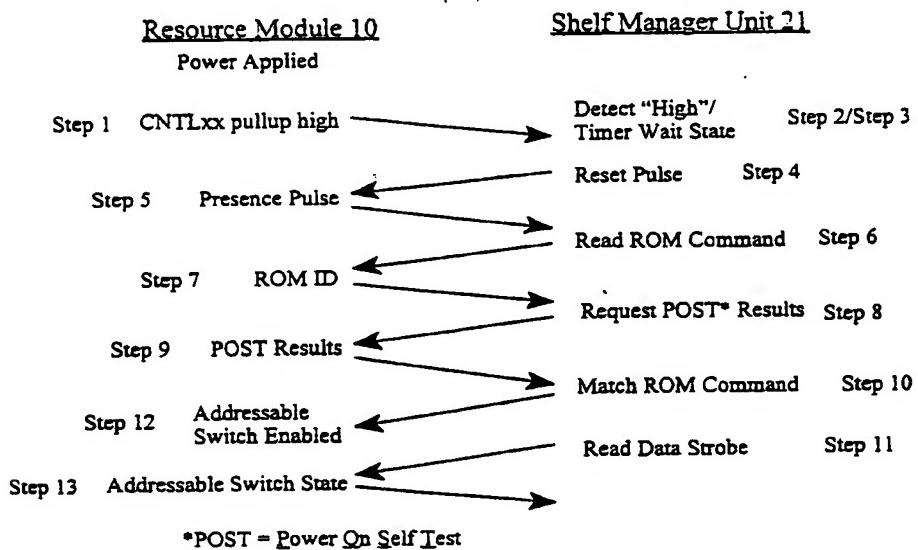


Chart 1. Grant Backplane Access

-89-

Step 1.

When power is applied, i.e. to an RM 10, CNTL xx is a 1-bit signal connected from each of the 1-25 slots in a shelf to the SM 21. The xx is replaced by the actual slot a resource module is inserted. For example, a resource module in slot 05 would control
5 CNTL05 as its 1-bit serial connection.

Insertion of the RM causes the 1-bit serial line CNTL xx to be pulled high.

Step 2.

The SM detects that a CNTL xx line has gone high. This tells the SM that a RM has been inserted.

10 **Step 3. Timer Wait State**

The fact that the RM signals its presence does not mean the processor in the RM has completed its download. The SM waits a minimum three seconds from receiving the CNTL xx signal to proceeding communication with that RM. This should allow the RM enough time to complete its POST.

15 **Step 4. Reset Pulse**

The shelf manager, upon timer expiration, initiates a reset pulse on the CNTL xx line. This is a “low” pulse that is held longer than 480 μ s. (See Ref. 15 for more details)

Step 5. Presence Pulse

20 The RM, upon detection of the reset pulse, initiates a presence pulse on the CNTL xx line. This is a “low” pulse lasting 60 μ s, \leq pulse \leq 240 μ s, (refer to the Dallas Semiconductor Automatic Identification Data book for more details). The presence pulse informs the SM that an addressable switch (AS) in the RM is alive and available.

25 **Step 6. Read ROM Command**

This command is sent form the SM to the RM addressable switch. This command requests a family code and serial number from the AS. The shelf manager

-90-

sends a 33h to the addressable switch, as specified in Reference 15. This protocol requires pulling the CNTLxx line low for 15 μ s, then placing a bit on the line. This is repeated until all eight bits or the ROM command are transferred.

Step 7. ROM ID

- 5 The RM provides its identification information as specified in Reference 15. This is a 64 bit sequence such that the first eight bits are the family code, the next 48 bits are a device unique serial number, followed by eight bits of CRC. The SM upon receiving this information verifies the family code and validates the CRC. The CRC polynomial is $x^8+x^5+x^4+1$.

10 Step 8. Request POST Results

The SM requests POST results from the RM. The SM transmits a 01101011 bit pattern, least significant bit first, on the CNTLxx line using a 100 μ s, pulse width for each bit. The RM monitors the CNTLxx line for the specified request POST results bit pattern.

15 Step 9. POST Results

- The RM responds to a SM POST results message with its POST results. This is a bit mask such that a one represents “pass” for a POST test and a zero represents a “fail” for a POST test. This allows the SM to know the results of each test. The RM first sends a 01101011 bit pattern, least significant bit first, to the SM as a POST result 20 preamble. This is followed by the POST results as represented in Table 16.

-91-

Table 16. POST Result Reporting

Bit #	POST Test	POST Result Values
1	Flash Memory Checksum Verification	Fail = 0 Pass =1
2	Random Access Memory Pattern Write/Read	Fail = 0 Pass =1
5	FPGA Done Bit	Fail = 0 Pass =1
4	Microprocessor and Timer Interrupt Test	Fail = 0 Pass =1
5	Bus Input/Output	Fail = 0 Pass =1
6	HDL Controller Test	Fail = 0 Pass =1
7	Resource Module Specific Test	Fail = 0 Pass =1*
10	Resource Module Specific Test	Fail = 0 Pass =1*
9	Resource Module Specific Test	Fail = 0 Pass =1*
10	Resource Module Specific Test	Fail = 0 Pass =1*
11	Resource Module Specific Test	Fail = 0 Pass =1*
12	Resource Module Specific Test	Fail = 0 Pass =1*
15	Resource Module Specific Test	Fail = 0 Pass =1*
13	Resource Module Specific Test	Fail = 0 Pass =1*
14	Resource Module Specific Test	Fail = 0 Pass =1*
15	Resource Module Specific Test	Fail = 0 Pass =1*

-92-

16	Resource Module Specific Test	Fail = 0 Pass = 1*
----	-------------------------------	-----------------------

*If a test is not defined, then return a 1

Step 10. Match ROM Command

This command is sent from the SM to the addressable switch in the RM (as specified in Reference 15). After verifying the ROM ID, a 55h followed by the ROM ID information is sent back to the addressable switch. The addressable switch receiving the 64 bit RIM ID sequence cause the device to enable, if currently disabled, or disable, if currently enabled. In other words this is used as an on/off switch for the addressable switch.

10 Step 11. Read Data Strobe

To verify that the resource module addressable switch is enabled, the SM performs a read data strobe on the CNTLxx line. This is defined as a “low” pulse lasting $1 \mu\text{s} \leq \text{pulse} \leq 15 \mu\text{s}$, (refer to Reference 15 for more details). If the CNTLxx line returns to a high state, then the addressable switch is enabled. If the CNTLxx line 15 remains in a low state, then the addressable switch is disabled.

Step 12. Addressable Switch State

Upon receiving a read data strobe, the addressable switch pulses the CNTLxx line “high” if it is enabled. This is defined as a high pulse lasting $0 \leq \text{pulse} \leq 45 \mu\text{s}$, (refer to the Dallas Semiconductor Automatic Identification Data book for more 20 details). If the addressable switch is disabled, then the CNTLxx line is pulsed “low”. This is defined as a low pulse lasting $0 \leq \text{pulse} \leq 45 \mu\text{s}$, (refer to Reference 15 for more details).

Step 13. Removing Module Backplane Access

The shelf manager, upon detection of a faulty module, should initiate a reset pulse 25 on the CNTLxx line. This is defined as a low pulse that is held longer than $480 \mu\text{s}$. Initiating a reset pulse should remove a faulty module from accessing the backplane.

-93-

Message Channel Startup Procedure

See Chart 2.

Step 1. Identify Module

The SM requests a RM to identify itself whenever a new RM has been inserted
5 or a shelf is being powered up. The SM sends the MSG_GET_S_RM_ID message.

Step 2.

The resource module responds with the MSG_ACK_S_RM_ID acknowledgment
message. The message status field states “OK” if the request was processed without
errors. The payload message has the rm_eqpt_attr information to identify the RM to the
10 SM.

Step 3.

Setting a resource module service state is initiated by the SM. The SM transmits
the MSG_SET_S_RM_PROV_OP_PROV_RM_SYS message to the RM requiring a
service state change. The rm_eqpt_attr payload is transported within the message with
15 the changed Primary Service State (PST).

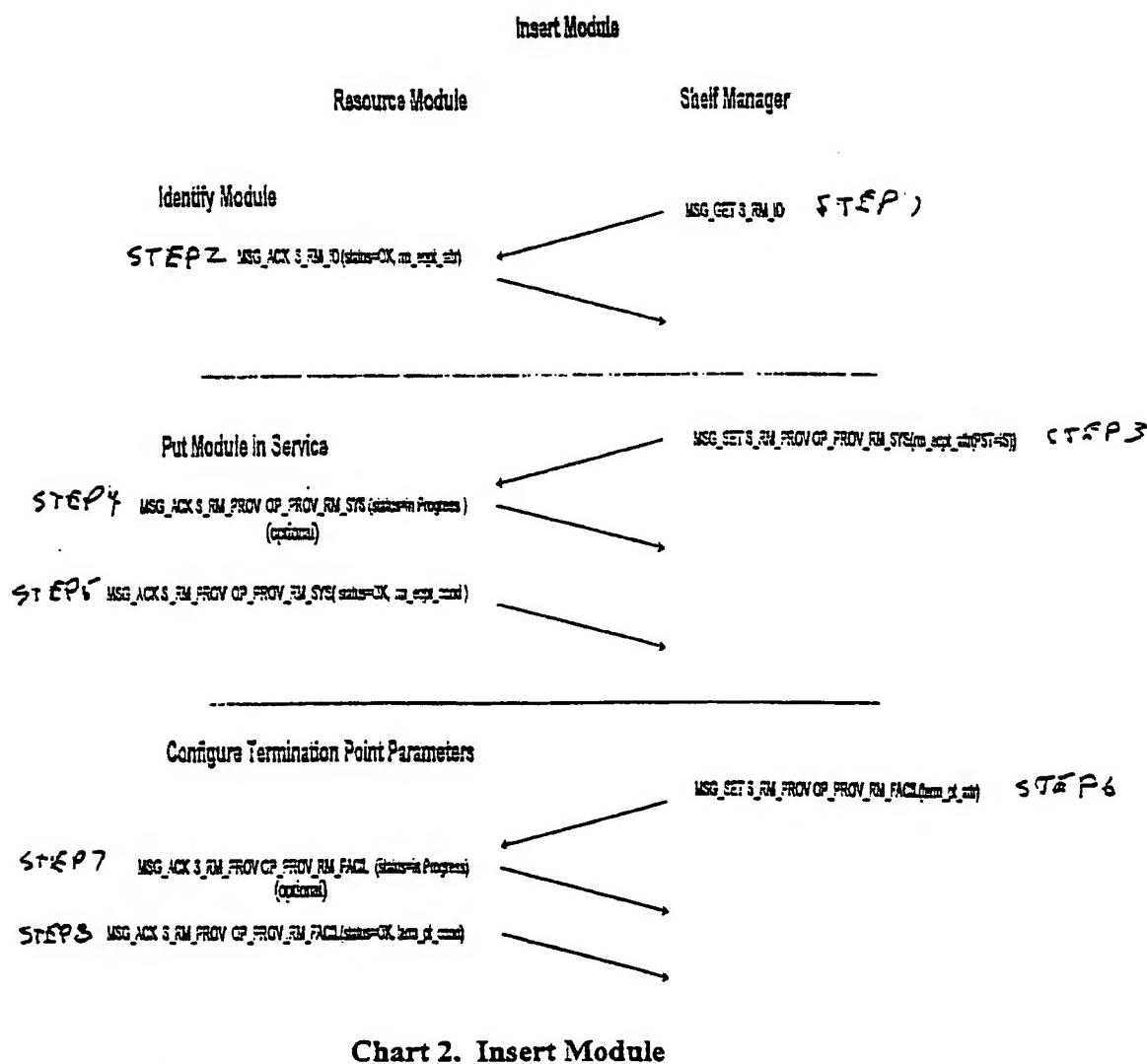
Step 4.

Optionally, the RM receiving a change primary service state message responds
with a MSG_ACK_S_RM_PROV_OP_PROV_RM_SYS acknowledgment message.
Returning a status of “In Progress” informs the SM that the message was received
20 without errors, but the message has not been completely processed.

Step 5.

When the RM has processed and implemented the changed primary
service state then the RM transmits the MSG_ACK_S_RM_PROV
OP_PROV_RM_SYS message. Returning a status of “OK” informs the SM that
25 processing of the changed primary service state is complete.

-94-



-95-

Step 6.

Configuring a RM is initiated by the SM. The SM transmits a MSG_SET_S_RM_PROV_OP_PROV_RM_FACIL message to the RM termination point requiring new/updated configuration parameters with the term_pt_attr payload. (Refer to the

- 5 Copper-LINC Object Model document for a detailed description of the term_pt_attr attributes.)

Step 7.

Optionally, the resource module receiving a configuration parameters message responds with a MSG_ACK S_RM_PROV_OP_PROV_RM_FACIL acknowledgment

- 10 message. Returning a status of "In Progress" informs the SM that the message was received without errors, but the message has not been completely processed within the resource module.

Step 8.

- When the RM has processed and implemented the updated configuration
 15 parameters then the RM transmits the MSG_ACK S_RM_PROV_OP_PROV_RM_FACIL message. Returning a status of "OK" informs the SM that processing of the new configuration parameters is complete.

Place Module in Service

- A diagram of the steps taken when a resource module is placed in service is
 20 shown in Chart 3.

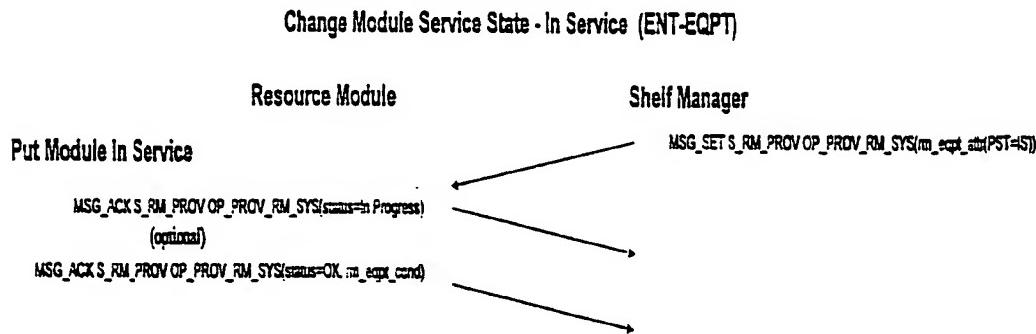


Chart 3. Change Module Service State - In Service

-96-

Step 1.

Setting a RM service state is controlled by the SM. The SM transmits a MSG_SET S_RM_PROV OP_PROV_RM_SYS message to each resource module requiring a service state change. The rm_eqpt_attr payload is transported within the
5 message with the changed PST.

Step 2.

Optionally, the RM receiving a change primary service state message responds with a MSG_ACK S_RM_PROV OP_PROV_RM_SYS acknowledgment message.
Returning a status of "In Progress" informs the shelf manager that the message was
10 received without errors, but the message has not been completely processed.

Step 3.

When the RM has processed and implemented the changed primary service state then the resource module transmits the MSG_ACK S_RM_PROV
OP_PROV_RM_SYS message. Returning a status of "OK" informs the shelf manager
15 that processing of the changed primary service state is complete.

This sequence is repeated for each module requiring a service state change.
RMs placed into service without a configuration parameters message operate in their
power-up default settings.

Change Module Service State

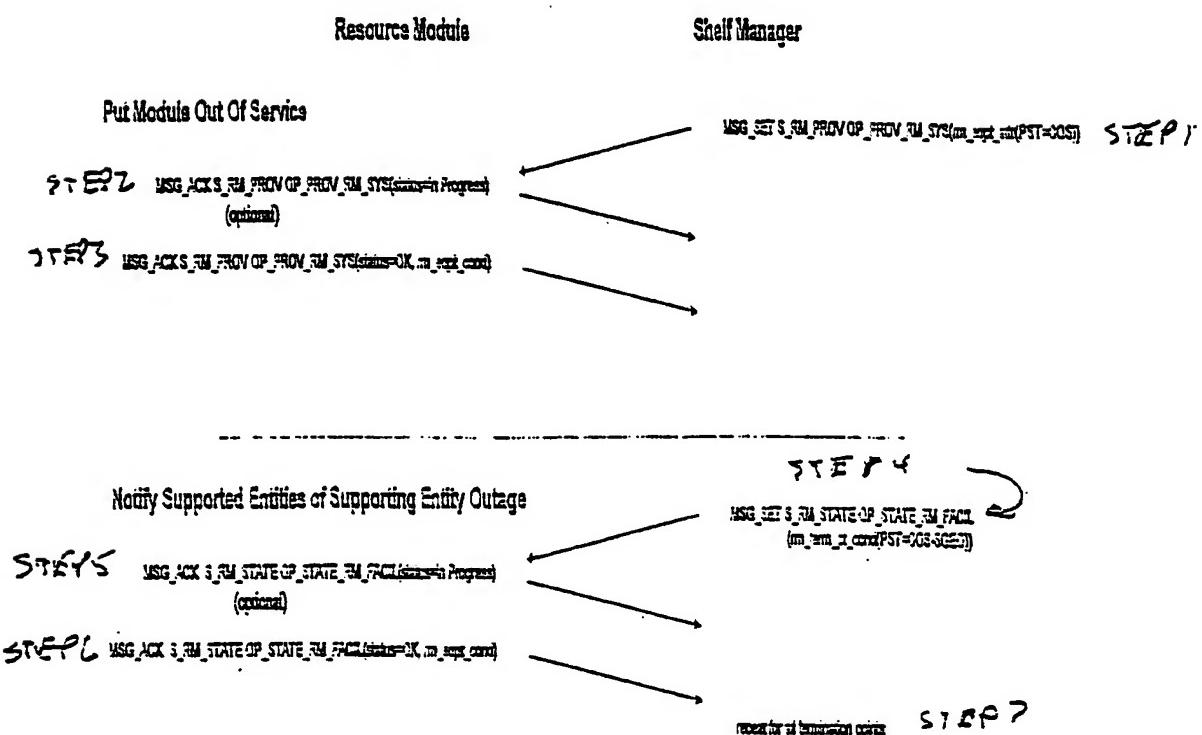
20 Chart 4 is a diagram of the steps that occur in a Change Module Service State -
Out of Service followed by an explanation of these steps.

Step 1.

Setting a RM service state is controlled by the SM. The SM transmits the MSG_SET S_RM_PROV OP_PROV_RM_SYS message to each RM requiring a
25 service state change. The rm_eqpt_attr payload is transported within the message with
the changed PST.

-97-

Change Module Service State - Out of Service (DELETE-SOPT)



Step 2.

- Optionally, the RM receiving a change primary service state message responds with a MSG_ACK_S_RM_PROV_OP_PROV_RM_SYS acknowledgment message. Returning a status of "In Progress" informs the shelf manager that the message was received without errors, but the message has not been completely processed.
- 5 received without errors, but the message has not been completely processed.

Step 3.

- When the RM has processed and implemented the changed primary service state then the RM transmits the MSG_ACK_S_RM_PROV_OP_PROV_RM_SYS message. Returning a status of "OK" informs the shelf manager that processing of the changed primary service state is complete.
- 10 primary service state is complete.

This sequence is repeated for each module requiring a service state change. RMs placed into service without a configuration parameters message operate in their power-up default settings.

Step 4.

- 15 Placing a RM out of service requires notification to the supported entities. The SM transmits the MSG_SET_S_RM_STATE_OP_STATE_RM_FACIL message to each RM termination point with the changed primary service state as part of the rm_term_pt_cond attributes.

Step 5.

- 20 The RM receiving a termination point primary service state message responds with a MSG_ACK_S_RM_STATE_OP_STATE_RM_FACIL acknowledgment message which informs the SM that the message was received without errors, but the message has not been completely processed.

Step 6.

- 25 When the RM has processed and implemented the changed termination point primary service state then the RM transmits the MSG_ACK_S_RM_STATE_OP_STATE_RM_FACIL message. Returning a status of "OK" informs the SM that processing of the changed primary service state is complete.

Change Termination Point Configuration

- 30 Chart 5. describes the steps that occur to change a resource module termination point parameter configuration.

-99-

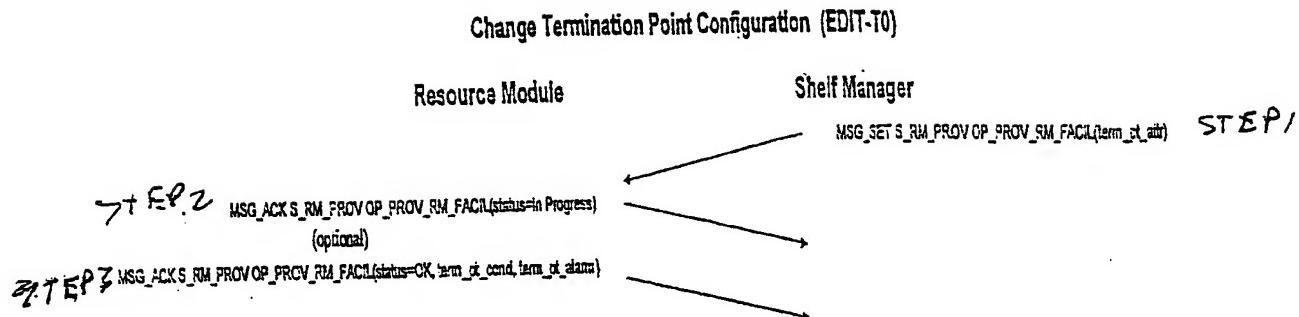


Chart 5. Change Termination Point Configuration

Step 1.

Changing a RM termination point configuration is initiated by the SM. The SM transmits the MSG_SET_S_RM_PROV OP_PROV_RM_FACIL message to each termination point requiring a change. The term_pt_attr payload is transported within the message.

Step 2.

The RM receiving a change configuration message optionally responds with a MSG_ACK_S_RM_PROV_OP_PROV_RM_FACIL acknowledgment message which informs the SM that the message was received without errors, but the message has not been completely processed.

Step 3.

When the RM has processed and implemented the changed configuration then the RM transmits the MSG_ACK_S_RM_PROV_OP_PROV_RM_FACIL message. Returning a status of “OK” informs the SM that processing of the changed configuration is complete. Included in the message the status of termination point conditions and alarms, as applicable.

This sequence is repeated for each termination point change.

-100-

Removing Modules

This section describes the steps required when an RM is removed from the system.

Physically removing a supported entity resource module is detected on the CNTLxx line

- 5 to the SM and internally the SM places that RM out of service as described in connection with Chart 4.

- 10 Physically removing a supporting entity RM has more system impact than removing a supported entity RM. If the supporting entity is the primary or secondary timing source, then the SM assigns a new timing source. Also, if the supporting entity is transporting supported entity data, then the supported entity RMs are placed out of service. Chart 6 is a diagram of the steps that occur.

Supporting Entity Resource Module Physically Removed

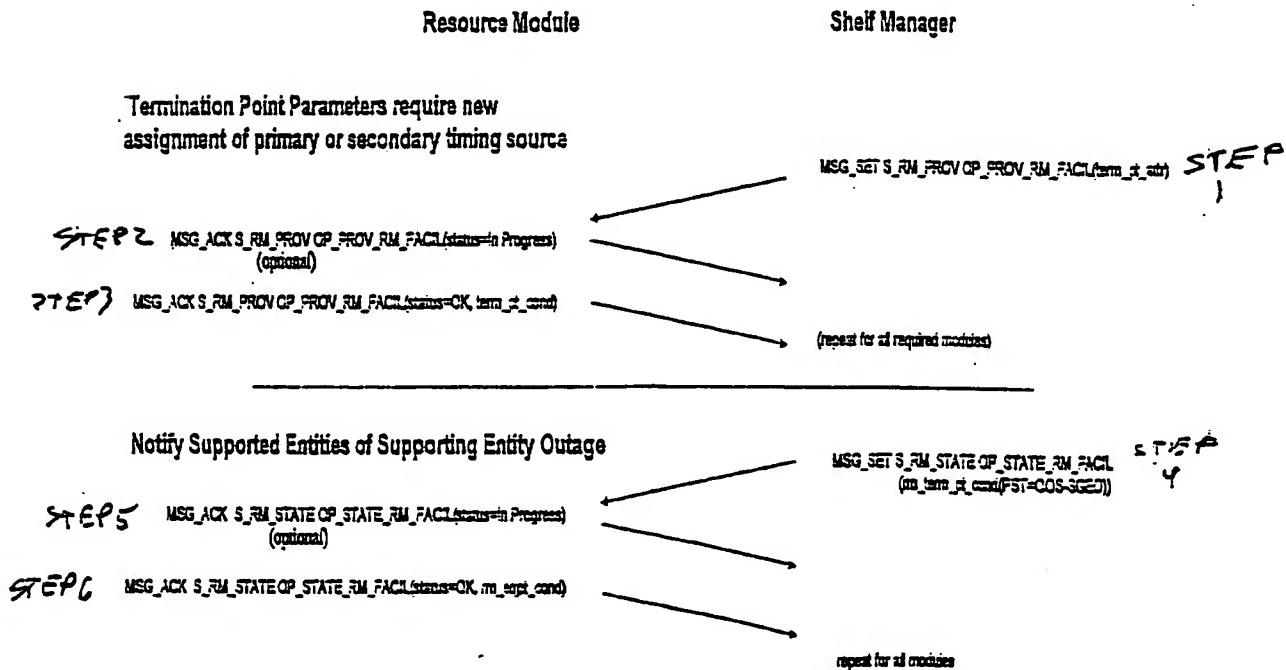


Chart 6. Supporting Entity Resource Module Physically Removed

-101-

Changing the Timing Source

Step 1

5 Changing the primary or secondary timing source requires transmitting a termination point parameters message by the SM. The SM transmits the MSG_SET S_RM_PROV_OP_PROV_RM_FACIL message to a RM termination point capable of being the timing source.

Step 2

10 The RM receiving a configuration parameters optionally responds with a MSG_ACK_S_RM_PROV_OP_PROV_RM_FACIL acknowledgement message which informs the SM that the message was received without errors, but the message has not been completely processed within the RM.

Step 3

15 When the RM has processed and implemented the updated configuration parameters then the RM transmits the MSG_ACK_S_RM_PROV_OP_PROV_RM_FACIL message. Returning a status of "OK" informs the SM that processing of the new configuration parameters is complete.

Supporting the Entity Outage Handling

20 Step 1 Removal of a supporting entity RM may require placing a supported entity RM out of service because of a supporting entity outage. The SM transmits the MSG_SET_S_RM_STATE_OP_STATE_RM_FACIL message to each RM termination point with the changed primary service state as part of the rm_term_pt_cond attributes. This causes the resource module to enter an Out of Service - Supporting Entity Outage (OOS-SGEO) state.

Step 2

25 The RM receiving a termination point primary service state message responds with a MSG_ACK_S_RM_STATE_OP_STATE_RM_FACIL acknowledgement message which informs the SM that the message was received without errors, but the message has not been completely processed.

-102-

Step 3

When the resource module has processed and implemented the changed termination point primary service state then the resource module transmits the MSG_ACK_S_RM_STATE OP_STATE_RM_FACIL message. Returning a status of "OK" informs the SM that processing of the changed primary service state is complete.

Report Alarm

RMs that can detect alarms autonomously report alarms to the SM. Autonomous reporting occurs, see Chart 7, when the alarm is set or cleared.

10

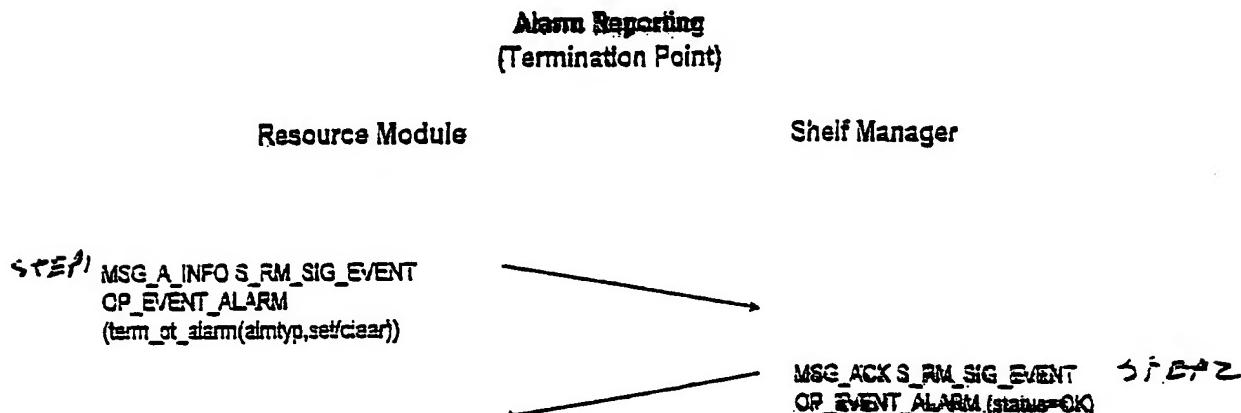


Chart 7. Alarm Reporting

Step 1.

When the RM termination point detects an alarm, that requires notification to the SM, the RM sends a MSG_A_INFO_S_RM_SIG_EVENT OP_EVENT_ALARM message, specifying the alarm type and a set or clear status.

Step 2.

The SM responds with a MSG_A_ACK_S_RM_SIG_EVENT OP_EVENT_ALARM message. Returning a status of "OK" informs the RM that the alarm message has been received and processed.

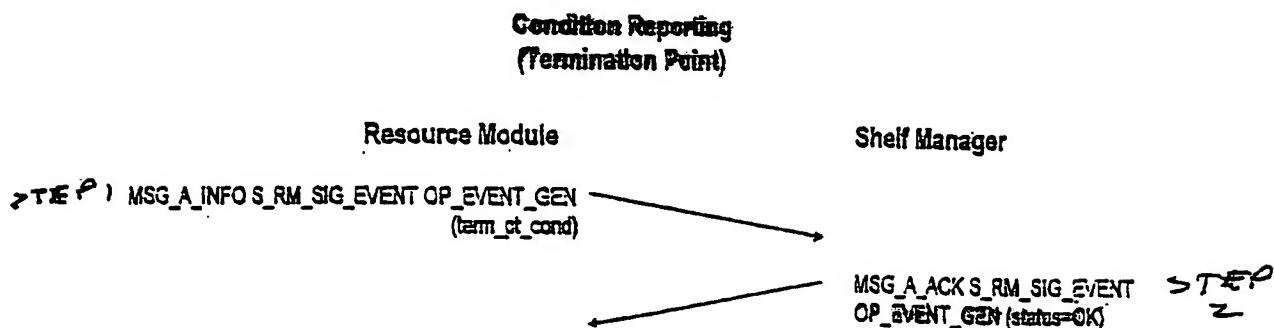
15

20

-103-

Report Condition

RMs that can detect conditions autonomously report conditions, described in Chart 8, to the SM.



5

Chart 8. Condition Reporting

Step 1.

The RM termination point detects a change in condition; the RM sends a `MSG_A_INFO_S_RM_SIG_EVENT OP_EVENT_GEN` message, specifying the condition code.

10

Step 2.

The SM responds with a `MSG_A_ACK_S_RM_SIG_EVENT OP_EVENT_GEN` message, which informs the RM that the condition message has been received and processed.

15

Request Status Information

The SM can also collect status information for RM equipment or an RM termination point. Chart 9 is a diagram of the steps taken to collect status information followed by an explanation of these actions.

-104-

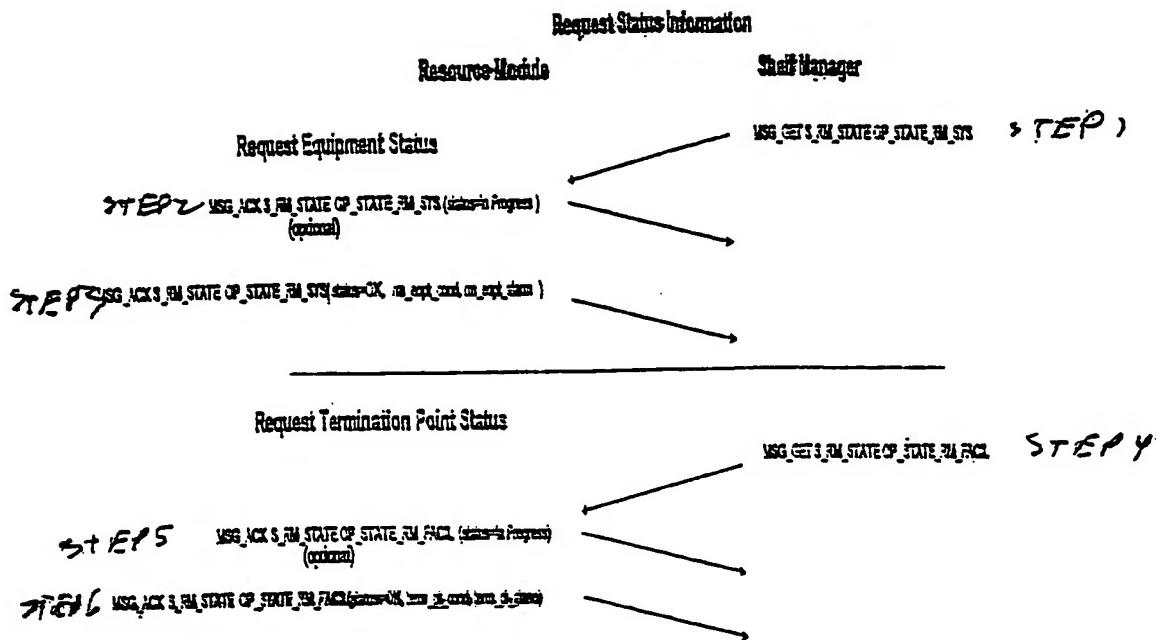


Chart 9. Request Status Information

Step 1.

5 The SM sends the `MSG_GET_S_RM_STATE_OP_STATE_RM_SYS` message to request equipment level status from a RM.

Step 2.

10 The RM responds with an optional `MSG_ACK_S_RM_STATE_OP_STATE_RM_SYS` acknowledgment message, which informs SM that the request has been received without transmission errors.

Step 3.

After processing the message the RM sends an `MSG_ACK_S_RM_STATE_OP_STATE_RM_SYS` message with a status of "OK", the `rm_eqpt_cond` and `rm_eqpt_alarm` attributes.

Step 4.

15 The SM sends the `MSG_GET_S_RM_STATE_OP_STATE_RM_FACIL` message to request termination point status from a RM.

-105-

Step 5.

The RM responds with the MSG_ACK_S_RM_STATE_OP_STATE_RM_FACIL message, which informs the SM that the request has been received without transmission errors.

5

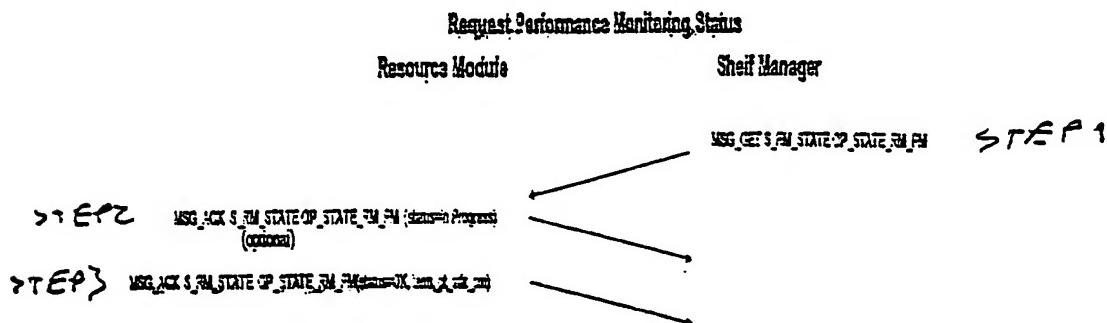
Step 6.

After processing the message the RM sends the MSG_ACK_S_RM_STATE_OP_STATE_RM_FACIL message with a status of "OK", the term_pt_cond, and the term_pt_alarm attributes.

10

Request Termination Point Performance Monitoring Status

The SM can also collect performance monitoring status for each RM termination point. Chart 10 describes the steps that occur to collect performance monitoring status.



15

Chart 10. Request Performance Monitoring Status

Step 1.

The SM sends the MSG_GET_S_RM_STATE_OP_STATE_RM_PM message to request termination point performance monitoring status from a RM.

Step 2.

20

The RM responds with the MSG_ACK_S_RM_STATE_OP_STATE_RM_PM message, which informs the SM that the request has been received without transmission errors.

-106-

Step 3.

After processing the message the RM sends the MSG_ACK_S_RM_STATE_OP_STATE_RM_PM message with a status of "OK" and the term_pt_attr_pm.

Keep Alive

5 The RM monitors traffic being sent to the SM. If the RM has not communicated to the SM within the last *three* seconds, then the RM sends a keep alive message, as shown in Chart 11, to the SM. This keep alive message informs the SM that the RM is still in the system operating.



10

Chart 11. Keep Alive

This message does not require an acknowledgment from the SM.

Maintenance

15 This section describes the actions that occur to operate a loopback. Chart 12 is a diagram of the steps that occur to operate a loopback.

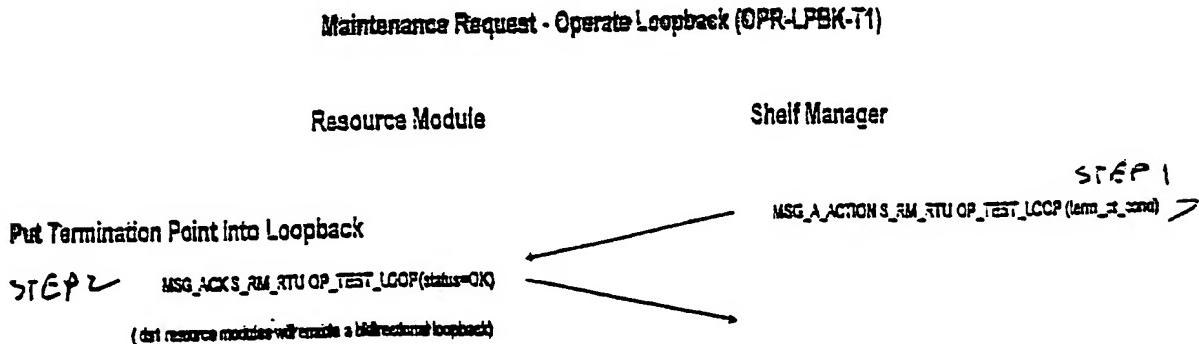


Chart 12. Maintenance Request - Operate Loopback

-107-

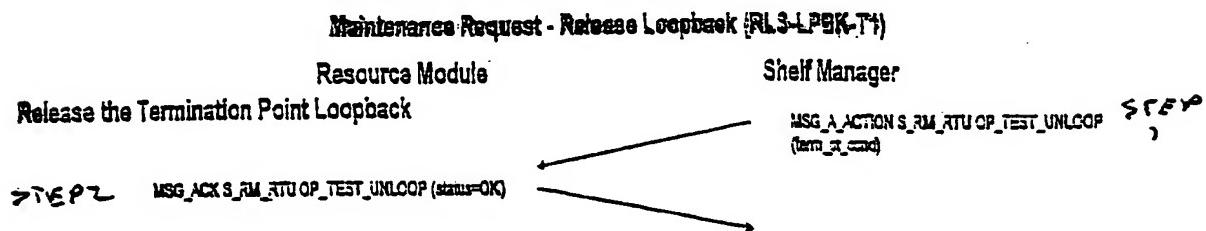
Step 1.

The SM sends the MSG_A_ACTION S_RM_RTU OP_TEST_LOOP message with the term_pt_cond attribute to request a DS1 loopback.

Step 2.

- 5 The RM responds with the MSG_ACK S_RM_RTU OP_TEST_LOOP acknowledgment message, which informs the SM that the request has been executed and a bi-directional loopback has been enabled.

To release a loopback see Chart 13.



10

Chart 13. Maintenance Request - Release Loopback

Step 1.

- 15 The SM sends the MSG_A_ACTION S_RM_RTU OP_TEST_UNLOOP message with the term_pt_cond attribute to release a DS1 loopback.

Step 2.

- The RM responds with the MSG_ACK S_RM_RTU OP_TEST_UNLOOP acknowledgment message, which informs the SM that the release has been executed and a bi-directional loopback has been removed.
- 20 The SM can also request an RM to perform a module self test. See Chart 14.

-108-

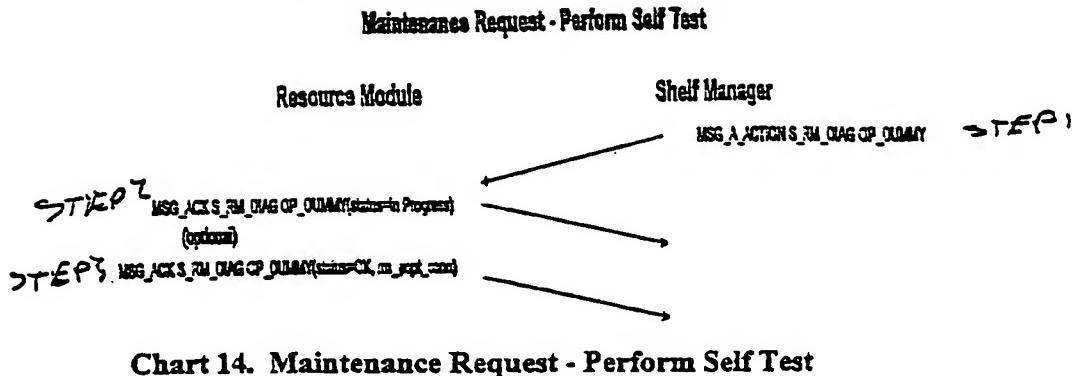


Chart 14. Maintenance Request - Perform Self Test

Step 1.

Commanding a RM to perform a self test is controlled by SM which transmits the MSG_A_ACTIONS_RM_DIAG_OP_DUMMY message to a RM.

5

Step 2.

The RM receiving a perform self test message responds with a MSG_ACKS_RM_DIAG_OP_DUMMY acknowledgment message, which informs the SM that the message was received without errors, but the message has not been completely processed.

10

Step 3.

When the RM has processed and executed the perform self test command then the RM transmits the MSG_ACKS_RM_DIAG_OP_DUMMY message, which informs the SM that the perform self test command is complete. Returning the rm_eqpt_cond attribute provides the results of the self test.

15

Software Upgrade

The LAS has the capability of software upgrade. This means the SM can transfer new executable code to a RM over the message channel, as shown in Chart 15, or over the time slot. This section describes the process of transferring the executable code over the message channel.

20

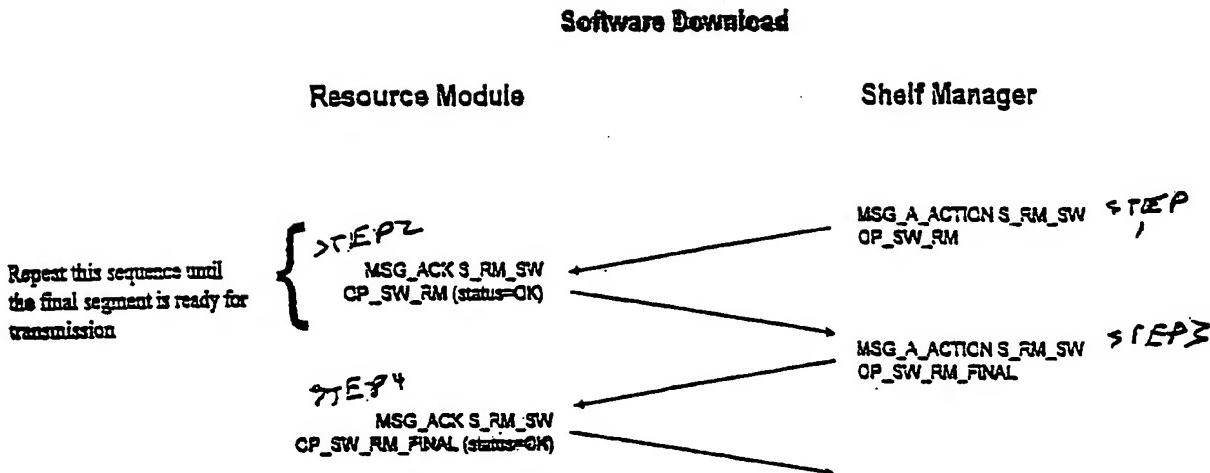


Chart 15. Software Download

Step 1.

5 The SM sends the MSG_A_ACTION S_RM_SW OP_SW_RM message with the new software as the payload.

Step 2.

10 When a RM receives, verifies, and stores the segment of new software then the RM sends the MSG_ACK S_RM_SW OP_SW_RM acknowledgment message, which informs the SM that the segment has been processed. This sequence is repeated until the last segment is prepared for transmission.

Step 3.

15 The last segment of software will be a MSG_A_ACTION S_RM_SW OP_SW_RM_FINAL message, informing the RM that this is the last segment.

Step 4.

When a RM receives, verifies, and stores this segment then the RM sends the MSG_ACK S_RM_SW OP_SW_RM_FINAL acknowledgment message, which informs the SM that the segment has been processed.

20 The messages described in the above section list the message acknowledgment status as either "In Progress" or "OK." Other reporting status'

-110-

exist if a message is transferred with errors. This section will provide details of the alternative acknowledgment status.

5 A module receiving a request that could not be executed because of other parameter settings within the module acknowledges the original message with a status of "Failed." The module that initiated the original message then attempts to resend the message twice before declaring a failure. This failure status has the highest priority because, for example, a bad crc calculation within a message could cause other failure status'.

10 A module receiving a request to modify a variable that cannot be changed returns a "Read Only" status in the acknowledgment message. The initiating module then logs this response and does not attempt to resend the message.

15 A module receiving a request to modify a variable with an out of range value returns a "Bad Value" status in the acknowledgment message. The initiating module logs this response and does not attempt to resend the message.

20 A module receiving an illegal message type returns an "Unknown Message" status in the acknowledgment message. The initiating module logs this response and does not attempt to resend the message.

25 A module receiving an illegal message subtype returns a "Bad Subtype" status in the acknowledgment message. The initiating module logs this response and does not attempt to resend the message.

A RM that receives a software download segment number that is out of order or is missing a segment responds with a "Bad Segment" status in the acknowledgment message. The SM then resends the first segment after the last known good segment transfer.

The following tables set forth a consolidated list of message channel messages:

-111-

Table 17. Shelf Manager Initiated Messages

Type	Value	Subtype	Value	Operation	Value	Attributes
MSG_SET	0x40	S_RM_PROV	0x04	OP_PROV_RM_SYS	0x11	rm_eqpt_attr
		S_RM_PROV	0x04	OP_PROV_RM_FACIL	0x10	term_pt_attr
		S_RM_STATE	0x19	OP_STATE_RM_FACIL	0x16	rm_term_pt_cond
MSG_GET	0x30	S_RM_ID	0x01	OP_DUMMY	0x00	
		S_RM_STATE	0x19	OP_STATE_RM_SYS	0x17	
		S_RM_STATE	0x19	OP_STATE_RM_FACIL	0x16	
		S_RM_STATE	0x19	OP_STATE_RM_PM	0x2B	
MSG_A_ACTION	0x50	S_RM_RTU	0x06	OP_TEST_LOOP	0x08	term_pt_cond
		S_RM_RTU	0x06	OP_TEST_UNLOOP	0x09	term_pt_cond
		S_RM_SW	0x03	OP_SW_RM	0x01	
		S_RM_SW	0x03	OP_SW_RM_FINAL	0x02	
		S_RM_DIAG	0x02	OP_DUMMY	0x00	
MSG_ACK	0x60	S_RM_SIG_EVENT	0x0A	OP_EVENT_ALARM	0x0A	status
		S_RM_SIG_EVENT	0x0A	OP_EVENT_GEN	0x0D	status

Table 18. Resource Module Initiated Messages

Type	Value	Subtype	Value	Operation	Value	Attributes
MSG_ACK	0x60	S_RM_ID	0x01	OP_DUMMY	0x00	status, rm_eqpt_attr
		S_RM_PROV	0x04	OP_PROV_RM_SYS	0x11	status, rm_eqpt_cond
		S_RM_PROV	0x04	OP_PROV_RM_FACIL	0x10	status, term_pt_cond, term_pt_alarm
		S_RM_STATE	0x19	OP_STATE_RM_SYS	0x17	status, rm_eqpt_cond, rm_eqpt_alarm
		S_RM_STATE	0x19	OP_STATE_RM_FACIL	0x16	status, term_pt_cond, term_pt_alarm
		S_RM_STATE	0x19	OP_STATE_RM_PM	0x2B	status, term_pt_pm
MSG_A_INFO	0x10	S_RM_SIG_EVENT	0x0A	OP_EVENT_ALARM	0x0A	term_pt_alarm(alarmtyp, set/clear)
		S_RM_SIG_EVENT	0x0A	OP_EVENT_GEN	0x0D	term_pt_cond
		S_RM_INFO	0xB	OP_WATCHDOG_RM	0x22	

Clock Synchronization

As noted, modules of the LAS 14 can be classified as a shelf managers (SM 21) or resource modules (RM 10). The SM type modules have the ability to master the system bus. The RM type modules have the ability to recover timing synchronization

- 5 information from a remote location or may require the ability to synchronize transmission with an external source.

The LAS clock interface 1400 (Fig. 47) serves two functions. The primary function of the clock interface is to provide a mechanism for the transfer of TDM data between modules within the LAS shelf. The second function is to maintain

- 10 synchronization between the telecommunications network and the telecommunications services deployed from the LAS 14. There are two elements of the clock interface. One element serves to master the timing for the entire system. The other element serves to provide timing information to the master for network synchronization. As shown in FIG. 47, the clock electronics is split into two regions. The upper region provides an
15 8KHz timing reference for network synchronization. The lower region provides system timing for the transfer of TDM data on the backplane. A Phase Locked Loop (PLL) 1420 shown in the lower half of the figure provides a 16.384MHz system backplane clock 408 and an 8KHz framing pulse 406. The system clock is split into four buffered clocks 402 for distribution on the backplane as shown by the four outputs SCLKX2N_1
20 through SCLKX2N_4. Each clock signal serves one of four Resource Module (RM) segments in the system. The bus switches provide a means to isolate the driving signals from the system backplane where there are multiple modules with the timing master interface as shown or in the case of live insertion into a functioning system. The interface design is such that it can act as a timing slave when another master is present
25 as indicated by the CLKFAIL signal 1404.

- The upper region of FIG. 47 shows the interface for network synchronization. Three modes of timing operation are supported: loop timing, office timing and internal timing are supported. Loop timing is provided by a recovered 8KHz clock signal which originates from an RM contained within the system. Such an RM must have the ability
30 to recover timing information from the loop. Not all RM types may have this capability. The LAS backplane provides three independent sources of a recovered

-113-

- 8KHz clock 410 which can be used as a loop timed synchronization source. In the Office Timed mode the timing master utilizes an interface to a Composite Clock (CC) source 1412 from a Central Office (BITS clock). As an option, a second interface to a composite clock source 416 may be present to support redundant CC sources. In
- 5 situations where Office Timing and Loop Timing sources are not present, an internally generated timing source can be supplied from a Stratum 4 oscillator 1414. Each of the mentioned 8KHz timing references can be assigned by switch 1418 as a Primary, Secondary or Tertiary source to maintain clock integrity in the event of a source failure.
- 10 The quality of the clock source is monitored by logic (not shown) to detect a failure.
- 10 Upon failure detection, the circuitry falls back to the next available reference source in priority order.

Office Synchronization (Fig. 48)

- The timing master provides an interface to the Composite Clock provided by a Central Office BITS. This timing source is a bipolar signal (Fig. 48A) which consists of
- 15 8 and 64KHz timing information. The timing master ensures the relationship between FSYNC (Fig. 48B) and the Office Clock. RMs that are timing slaves ensure the relationship of the Byte (Fig. 48C) and Bit (Fig. 48D) (8 & 64KHz) clocks as shown. Because of this timing relationship, the need to transmit the Byte and Bit clock on the backplane is eliminated.

20

Recovered Clock Provisions (Fig. 49)

- Each RM 10 that has the ability to recover timing information from its telecommunications loop interface implements the circuitry shown in FIG. 49. The RM provides the means to supply one recovered clock from any number of loop timed
- 25 sources 1432 to any of three backplane interface signals 1430. The three backplane signals 1430 are used by the system as the primary P, secondary S and tertiary T sources of loop timing.

-114-

Slave Operation (Fig. 50)

RMs 10 which do not have the ability to master the system timing as described above must recover timing information from the backplane as shown in FIGs. 50 and 51. In this type of module the TDM interface 1450 requires the local generation of a 5 clock signal called SCLK. The module may also require the local generation of 8 and 64KHz clocks 1451, 1452 for DS0 type interfaces as shown in FIGs. 50 and 51.

Clock Handoff

The timing interface allows for two modules in the system to master the timing. Only one module at any given time may be the timing master and only cards in slots 25 10 & 26 have the ability to provide the necessary timing signals to the other RM's. A normal TDM framing cycle is shown in FIG. 52. Data is transmitted in bit serial fashion and framed by the FSYNC signal. This pattern allows 128 bytes of information to be transported by the backplane. Under normal conditions the CLKFAIL line is driven low by the active master at all times. The CLKFAIL signal is an open collector 15 output of the active master. All potential timing masters pull this signal high. The lack of drive by the current master creates a high condition and indicates a failure. A failed condition would trigger an uncontrolled handoff to an armed master. This type of transfer would result in a brief loss of service as there would be no timing information until the new master takes control. The CLKFAIL signal is also used to synchronize the 20 transfer of timing mastership to another master in a controlled manner as shown in FIG. 53. As shown, a coordinated handoff results when the CLKFAIL signal is driven high during the low portion of the FSYNC pulse. Immediately after, the current master "parks" the output clock signals in a high state and then transitions into a high impedance state. This allows an alternate master to begin the next TDM frame cycle in 25 the correct sequence which does not interrupt service.

Shelf Manager Timing

The timing and clock synchronization functionality embodied in a shelf manager is now described.

Timing Signal Inputs

Composite Clock Input

- The SM recovers a composite clock synchronization signal supplied externally to the system. Composite clock is a synchronization signal that is a 64 kbps, 5/8 duty cycle, all-ones, bipolar return to zero signal with a bipolar violation every eighth pulse. The input composite clock preferably meets the requirements of Bellcore TA-TSY-000378 and ANSI T1.101. In particular, the SM recovers a primary composite clock synchronization signal supplied externally to the system. The SM can also recover a second composite clock synchronization signal supplied externally to the system. The second composite clock source can be used as a potential clock fall back upon fault detection of the primary composite clock source when the timing mode is office timed.

Recovered 8KHz Clock Inputs

- The SM has the ability to select from three independently recovered 8KHz clock synchronization sources. These recovered 8KHz clock sources are supplied by RMs with the ability to recover timing from a remote source. These clock sources are available from the system backplane and meet the requirements of a standard TTL level signal. The recovered 8KHz backplane signal is susceptible to reflections which may cause false triggering if used as a clocking signal. The incident waveform transition indicates the true timing reference point. Subsequent transitions which may occur within a 100nS time period are ignored.

Clock Fail Signal Input

- The SM has the ability to monitor the status of the CLKFAIL backplane signal. This input is used to detect the absence of a system bus master or as a trigger to initiate a controlled system bus master handoff from the current master to an armed redundant master.

Timing Signal Outputs

A frame sync output is provided as FSYNC on the system backplane. This output can be synchronized with an enabled timing synchronization source input signal. System clock outputs are provided as SCLKX2N_n (where n = 1 to 4) on the system backplane. This output can be synchronized with the enabled timing synchronization source input signal. The SM has the ability to drive a logic low on the CLKFAIL backplane signal. The CLKFAIL output is used to signal the presence of a system bus master or as a trigger to initiate a controlled system bus master handoff from the current master to an armed redundant master.

10 Clock Monitoring (Fig. 47)

Each timing synchronization source includes circuitry 1460 to enable monitoring of the health of the source. Such circuitry is sufficient to detect clock impairment and allow management of the clock recovery process.

Timing Modes

15 The user has the ability to select the following timing modes for system synchronization of the outputs defined above. The selection of timing synchronization mode is determined by selection through an Operator Interface or any capable Operations Support System interface. In the office timing mode, the primary system synchronization source is traceable to a composite clock input signal which is supplied externally to the system. In loop timing mode, the primary system synchronization source is traceable to a recovered clock input signal. The SM provisions RMs which are capable of clock recovery to provide a recovered clock source on each of the recovered clock inputs. In internal timing mode, the primary system synchronization source is traceable to a locally generated timing source. The internal timing source preferably meets stratum 4 requirements, as specified in TA-NWT-1244, Section 3.

Clock Holdover Timing

Clock holdover is defined as the ability to maintain the phase and frequency relationship of the output clocks with respect to the timing reference when the timing

-117-

reference is absent. The holdover state is intended to maintain the system timing relationships for a momentary period of time until an alternate timing source is supplied or the current timing source restored.

- The loss of the current timing source without an immediately available alternate
- 5 source causes entry into the clock holdover state within 100mS. Entry into holdover causes no more than 500 nsec variation in the time delay between the leading edges of the heldover timing signal and the DS0 output data. During the initial 15 seconds of holdover, there is preferably less than 500 nsec. of phase-time movement on the internal clocks. If the external CC reference is restored within the period where the DS0 data is
- 10 still phase aligned, the recovery preferably causes no more than one erred second. If the external CC reference is restored past the period where the DS0 data is no longer phase aligned, restoration of the DS0 physical layer that is phase aligned to the external composite clock signal preferably occurs within one second.

User Selected Synchronization Options

- 15 The user has the ability to select a primary synchronization source from the timing modes described above at all times. The user can select a secondary synchronization source from those timing modes omitting the office timing source. The selection of a secondary source is only allowed after the selection of a primary source. The user can select a tertiary synchronization source from those timing modes omitting
- 20 the office timing source. The selection of a tertiary source is only allowed after the selection of a secondary source.

Clock Fallback & Reversion

- Clock fallback is the automatic switching from the user selected synchronization timing source to an alternate timing source upon the detection of a clock fault. Clock
- 25 reversion is defined as the ability for a previously failed timing synchronization source to be restored as a source upon signal recovery.

The fall back scheme includes three possible synchronization sources designated as primary, secondary and tertiary. A fourth internally generated source is available at all times. The four possible synchronization sources comprise the synchronization

-118-

source table. Clock fallback and reversion is accomplished in a round robin manner. Round-robin is defined as reversion to the top of the synchronization source table and occurs only when the set of timing synchronization sources in the table have been exhausted. The current fallback state and synchronization source are obtainable from 5 the user interface. The source status (impaired or unimpaired) is also available for each user selected source. The rate of phase change of the clock signals when switching synchronization sources is limited according to TA-NWT-1244.

The state diagram of FIG. 54 is applied upon impairment of the current synchronization source. Switching occurs within 100 milliseconds from when 10 impairment is detected. The synchronization source is declared impaired by detection of a Loss of Signal (LOS) condition or detection of a fractional frequency offset exceeding $1.1 \times (\text{accuracy} + \text{pull-in range})$. For stratum 4 this is $1.1 \times (32 \times 10^{-6} + 32 \times 10^{-6}) = 70.4 \text{ PPM}$ or a 0.5632 Hz offset in an 8KHz reference. Loss of Signal is defined as one or more missing clock pulses. The LOS condition no longer exists when 15 a correct clock is received for a minimum of 1 millisecond. The Composite Clock (CC) is declared impaired by detection the criteria defined above in addition to an incorrect bipolar violation density that departs from the one in eight pattern.

The state diagram of FIG. 54 is applied upon recovery of a previously impaired synchronization source. Reversion from holdover to normal mode when Office timed is 20 allowed following the detection of valid Composite Clock for a minimum of 10 seconds.

Clock Handoff

Clock handoff is the switching of system bus mastership, and hence system timing synchronization, from the current (active) master to an alternate master. The SM 25 serving as the active bus master is indicated by a lighted "In-Service" LED on the faceplate. Only one such unit shall be in the "In-Service" state at any given time. The SM in the "Out of Service" state can be removed from the shelf without inducing any system errors. Uncontrolled handoff takes place in the unplanned failure or removal of the active bus master. The removal of the SM serving as the active bus master without 30 administratively placing it in the "Out-of-Service" state causes bus mastership to be

-119-

transferred to an alternate bus master. Preferably no more than one erred second on any DS0 occurs due to this transition. Removal of the active master without an alternate master could cause a system wide outage of service. Controlled handoff is the planned handoff of bus mastership from the active master to an alternate by user driven

- 5 command. The SM serving as the active bus master transfers bus and timing mastership to an alternate unit upon being commanded to an "Out-of-Service" state from its Operator Interface. This action causes the alternate unit to be placed in the "In-Service" state. The SM serving as the alternate bus master transfers bus and timing mastership from the active unit upon being commanded to an "In-Service" state from its Operator
- 10 Interface. This action causes the active unit to be placed in the "Out-of-Service" state. The command to cause the clock handoff is inhibited if an alternate SM is not present or not capable of performing the bus master function. The user is preferably notified in such instances. The transition to an alternate bus master should result in no clock aberrations. Worst case, no more than one erred second on any DS0 interface results.

15 Timing Quality

The SM preferably complies with timing synchronization requirements specified for equipment employing DS0 interfaces. These requirements are shown in Table 38. For interpretation of unit requirements the leading edge of the DS0 output signal has the same meaning as the leading edge of the 8KHz FSYNCN output signal.

-120-

Table 38. Timing Synchronization Requirements

Requirement	Specification	Notes
Phase Alignment	0 - 1 microsecond	The time delay between leading edge of the composite clock input signal and the leading edge of the DS0 output signal
5 Wander	<100 ns	For low frequency jitter (wander) for up to 10 Hz on the composite clock leading edge input the DS0 output signal leading edge shall track within the number specified.
Phase Hits	500 ns pk - pk	For phase hits on the composite clock input of up to 1 ms, with a maximum rate of change of 81 ns in 1.326 ms, the DS0 output leading edge shall track the composite clock input leading within the number specified.

The unit preferably is capable of meeting the Stratum 3 accuracy, stability and pull in range requirements specified in TA-NWT-001244. Those and the composite clock phase alignment objectives are shown in Table 39:

-121-

Table 39. Timing Synchronization Objectives

Requirement	Specification	Notes
Minimum Accuracy	4.6 ppm	Long term deviation from the nominal frequency without external reference
Phase Alignment	250 ± 50 ns	The time delay between leading edge of the composite clock input signal and the leading edge of the DS0 output signal
5 Stability	$\pm 3.7 \times 10^{-7}$ per day for the initial 24 hours	The rate of change of frequency from the input frequency once the external reference is removed.
	[20×10^{-9} (initial) + 355×10^{-9} (due to temperature) + 10×10^{-9} (for miscellaneous)]	Stability with temperature components factored in.
Wander Generation	sec. 5.3 TA-NWT-1244	
Wander Transfer	sec 5.4 TA-NWT-1244	
Jitter Generation	sec 5.5 TA-NWT-1244	

10 Clock Related Alarms

A minor alarm is declared upon loss of one or more reference sources (e.g. Composite Clock if office timed or the Primary if loop timed) or the detection of a fractional frequency offset exceeding $1.1 \times$ (accuracy + pull-in range), for stratum 4 this is $1.1 \times (32 \times 10^{-6} + 32 \times 10^{-6}) = 70.4$ PPM or a .5632 Hz offset in a stratum 4, 8KHz reference. A major alarm is declared if a minor clock alarm condition persists for 24 hours or upon loss of all reference sources with Fallback to Internal Timing.

-122-

Resource Module Timing

The timing and clock synchronization functionality embodied in a resource module is now described.

Timing Signal Inputs

- 5 A Frame Sync Input signal is provided as FSYNC on the system backplane. This input is synchronized with the enabled timing synchronization source signal provided by the SM. A System Clock Input signal is provided as SCLKX2N_n (where n = 1 to 4) on the system backplane dependent upon the particular slot address the RM occupies. The local version of the signal is labeled SCLKX2N. This input is
10 synchronized with the enabled timing synchronization source signal provided by the SM. The RM has the ability to monitor the status of the CLKFAIL backplane signal. This input is used to detect the absence of a system bus master or as a trigger to initiate a controlled system bus master handoff from the current master to an armed redundant master.

15 Timing Signal Outputs

- An RM can recover 8KHz timing synchronization information from a remote source. The RM can drive REC8KHZ_P, REC8KHZ_S, and REC8KHZ_T output backplane signals with the recovered clock source upon provisioning command from the SM. The RM presents a high impedance on the respective signals otherwise. In the
20 event the RM contains multiple recovered sources, the specific source is also selectable via provisioning command from the SM.

External Synchronization And Timing Recovery

- For an RM that requires DS0 synchronization and timing information, an 8KHz Byte Clock Recovery signal is regenerated by using the SCLKX2N and FSYNCN
25 timing signals. These two input timing sources are synchronized to an external composite clock source by the SM. A 64KHz Bit Clock Recovery signal is regenerated

-123-

by using the SCLKX2N and FSYNCN timing signals. These two input timing sources are synchronized to an external composite clock source by the SM.

Circuit Card Hot-Swap System for LAS

- The hot-swap system allows resource modules (Rms 10) of the LAS 14 to be inserted and removed from the system without inducing errors or interrupting service provided by other modules in the system. The hot-swap system is divided into three elements; electrical, mechanical and software.

Electrical Element (Figs. 55 and 56)

- The electrical element includes circuitry which serves to protect a time division multiplex (TDM) bus 42 of the LAS 14 that is used to route communications data between modules. Protection of this bus is essential to error free operation of other modules in the system. These two circuit elements are known as the 'inrush current limiter' 606 and 'bus isolation' circuit 600. A diagram of the bus isolation circuitry is shown in Figure 55.

- The electrical element further includes circuitry that limits the in-rush current upon module insertion. This protects the primary power distribution system from disruptions that could interfere with proper module operation. A block diagram of the in-rush circuit is shown in Figure 56.

RM Bus Isolation

- Each RM 10 includes a mechanism that allows the shelf manager SM 21 to isolate a malfunctioning RM from the backplane 42. Insertion or removal of the RM does not cause transmission errors on timeslots in use by other cards due to the bus isolation mechanism. The bus isolation mechanism comprises precharge voltage, isolation modules, initialization request, bus enabling and sanity test as follows:

-124-

Precharge Voltage and Isolation Modules

A +5V precharge available on all slots of the LAS can be used to provide early make/late break connections on +5V for the purposes of providing power to bus isolation modules 600. These modules include high speed CMOS bus switches 602

5 designed to provide hot swap capability. The devices are designed to power up in the high impedance state. This isolates the module backplane from any TDM bussed signals, clocks and Message Channel. The bus isolation modules are controlled by addressable switches 602 that in turn are under the control of the SM 21. The addressable switch powers up so that I/O is inhibited, and remains so until addressed

10 and instructed by the SM.

Bus Switch

The bus switches 602 are designed to be inherently high impedance when the circuit card has no power or is actively disabled. A Dallas switch DS2405 or DS2407, which is controlled by a backplane signal CNTLi enables or disables the

15 message channel from the backplane. The Dallas switch provides each circuit card with a unique serial number.

The DS2405 device provides one switch which controls operation of the message channel isolation switches. An optional DS2407 device provides two switches, the second of which is used to disable the on board power converter and remove the

20 circuit card load from the main power bus. When the Dallas switch is disabled, the TDM Bus 42 is disconnected from the backplane. When the Dallas switch is enabled, the TDM Bus and the clocks switches are controlled by the onboard microcontroller 604.

Initialization Request

25 Upon insertion, a RM 10 requests initialization by pulling the CNTLi ($i = 1$ to 25) lead high. Figure 57 shows a timing diagram for an initialization request. This lead is used to signal the SM that a module has been inserted. The SM responds by transmitting a reset pulse on CNTLi. The addressable switch 602 on the module responds by transmitting a presence pulse. Once the presence pulse has been detected,

-125-

- the SM interrogates the addressable switch for its factory assigned unique address, then requests that the RM report the results of its power on self test (POST). If the results are satisfactory, the SM then proceeds to enable the RM backplane I/O. This will place the bus isolation modules 600 into the transparent state. The module can be allowed to
- 5 communicate on the backplane when the shelf manager has interrogated the module's addressable switch using CNTLi. The "one-wire protocol" is used for such requests.

- Interrogation for the unique address consists of issuing an 8 bit Read ROM (0x33) command to the addressable switch on the module. The CNTLi lead incorporates the standard Dallas Semiconductor 1 Wire protocol for data transfers, with
- 10 all data being read and written least significant bit first. The module's addressable switch responds with the contents of its 64 bit ROM according to the following format:

MSB LSB	MSB LSB	MSB LSB
8 bit CRC code	48 bit serial number	8 bit family code (0x05)

- Once the SM has determined the exact 64 bit ROM sequence of the target
- 15 switch, it will transmit an 8 bit Match ROM (0x55) command followed by the 64 bit ROM sequence. This causes the addressable switch to toggle its I/O control ON.

Sanity Test

- After a RM is in service, there exists the possibility that a malfunction could cause it to impact other modules by causing errors on the bus. In order to minimize the
- 20 impact that a malfunctioning module may have, the module provides a watchdog function that will cause the CNTLi lead to be pulsed low within 600ms for a duration of 100ms. The SM deems the absence of a watchdog pulse within the 600ms window as an indication of a malfunctioning unit. The SM causes the malfunctioning module to be disabled by pulling and holding the CNTLi lead low (minimum of 480 ms). This
- 25 effectively removes power from the addressable switch 602 returning its I/O control to the default state of OFF.

After the malfunctioning module has been isolated from the bus, the SM periodically (every 500ms) releases the CNTLi lead (for 200 ms) to determine if the

-126-

module has been physically removed from the slot. Releasing the CNTLi lead at this point does not impact the module access to the backplane since I/O remains OFF. If the module has been removed, the SM then looks for insertion of a new module. If the module has not been removed, the SM continues to look for module removal.

- 5 Alternatively, a module which is actively connected to the backplane can be physically removed. Upon detecting that the CNTLi lead is no longer pulled high (minimum of 480 ms), the SM interprets this as a module removal and looks for the CNTLi to be pulled high when another module is inserted into that slot. The RM provides a watchdog function on the CNTLi lead by pulsing the line low within every
10 600 ms, for 100 ms. The RMs tri-state the backplane I/O if the CNTLi lead is held low for > 480 ms.

In addition to providing the watchdog function on the CNTLi lead, the RM periodically transmits a "keepalive" message. The watchdog function provides a lower level indication of the RM status that can be transmitted and monitored with a relatively
15 high frequency reducing the time that common resources are exposed to an RM failure. The keepalive message provides a more comprehensive method of verifying the health and status of an RM but at a much lower frequency, conserving message channel bandwidth.

Mechanical Element (Figs. 58-61)

- 20 The mechanical element of the hot-swap technology consists of guide pins which serve to align plug in modules and assure that an electrical ground connection is established prior to any other electrical circuit connection. The guide pin arrangement consists of male pins 630 on the backplane connector 632 which mate with female pin sockets 634 on the plug in module. The backplane guide pin 630 is shown in Figure 58.

25 Software Element

The software element of the hot-swap technology ensures protection of the backplane TDM bus upon module insertion or failure. Details of the backplane access protocol have been described previously in connection with the description of the One-Wire Protocol and Message Channel.

-127-

2SDSL

The following describes an embodiment of an SDSL resource module having two SDSL line interfaces. The 2SDSL Module is a resource module 10 in the LAS 14 (Fig. 6) intended for use in transport of 2B1Q formatted data at rates that correspond to 5 4, 6, 8, 12, 18, 24, 30, 32 and 36 DS0s. The rates are configurable through the Operator Interface hosted on the Shelf Manager Unit 21. Also included is a 16kbps framing channel that supports an embedded operations command (EOC) channel.

The main applications for the 2SDSL module are for use with Integrated Access Devices and for Backplane Extension as described in the following sections. Also 10 described are the signaling behavior and transport of POTS and data circuits through the 2SDSL RM.

Fig. 60 illustrates an Integrated Access Device (IAD) application. In this application, a 2SDSL module 820 is used to provide inexpensive backhaul for a customer's LAN and POTS voice services at premises 824 to networks 810, 812 over 15 SDSL lines 821. The figure displays IADs 822A, 822B connected to a single 2SDSL resource module.

In a backplane extension application as shown in Fig. 61, LAS shelves 12 are connected together in a "daisy chain" fashion--essentially concatenating the backplanes of all the connected frames. In other applications, a 2SDSL Module can be used in the 20 transport of frame relay data and for POTS concentration for GR-303.

The 2SDSL RM can be configured to support three different types of payload timeslot structures for communication with an IAD over an SDSL line within a 36-DS0 frame:

- IAD frame data with voice
- IAD frame data without voice
- clear channel data frame

The IAD frames support CPE 822A, 822B (Fig. 60). The clear channel frame is designed to support connections between two shelves (Fig. 61). Figs. 62A-62C display the three payload timeslot structures. A Signaling DS0 832 transports voice signaling 30 information for POTS channels 834 to and from an IAD, and a Reserved DS0 830

-128-

supports OAM&P for any data channels 836 connected to an IAD. For shelf-to-shelf configurations, neither the Reserved or Signaling DS0 channels are required, and a "clear" digital channel with 36 DS0s can be supported.

- For communication between the 2SDSL resource module 820 and an IAD 822,
- 5 the transport mechanism can be either PCM, ATM, high-level data protocol (HDLC), point-to-point protocol (PPP), Frame Relay, or a combination of these on a TDM structure. The types of data being transported range from real-time voice to bursty packetized data.

The IAD 822A (Fig. 60) includes 4 POTS ports and a 4-port Ethernet hub

10 supporting TCP/IP as well as various routing functions. The Transport is TDM for both voice and data (Fig. 62A) with voice using PCM and data using HDLC encapsulation or PPP. In an embodiment, the four POTS interfaces provided at the IAD 822A are managed by the SM 21 (Fig. 6). Once the SDSL transport path is In Service, the POTS channels are available to be placed into service.

15 The IAD 822A has the responsibility of analog line signaling and supervision, as well as digital signaling bit insertion/extraction. The POTS interfaces are provisionable as Universal Voice Grade or Single Party "channel units". State management (such as ON-HOOK and OFF-HOOK) is performed at the IAD and communicated to the 2SDSL RM. The RM, in turn, provides proxy POTS functionality and forwards commands

20 from the SM on to the IAD and indications from the IAD on to the SM. Raw signaling, however, is passed through transparently.

Voice Signaling is via the dedicated DS0 832 (Fig. 62A) with up to 24 channels multiplexed in both the upstream and downstream directions with provisions for future expansion to 30 channels per DS0. On the downstream side, logic in the 2SDSL 820 is

25 capable of monitoring for various signaling bit patterns, and setting discrete outputs upon detection. On the upstream side, the logic allows various signaling bit patterns corresponding to the detection of certain discrete input states. Signaling bit interpretation can be provisionable as either D4 (superframe), TR08, ESF, or GR303 for each line interface.

30 Trunk conditioning that occurs in-band to the voice/signaling DS0s is handled at the IAD 822A. Trunk conditioning initiated in-shelf (due to SM command) is acted on

-129-

by the SDSL RM with the appropriate tone inserted and the proper analog signaling states determined by the SDSL RM and communicated to the IAD 822A via the EOC.

A data interface is provided via the remaining DS0s not reserved or used for signaling/voice and will vary in bandwidth according to the line rate. The 2SDSL RM
5 provides a transparent data path for the assigned data DS0s. Once the SDSL transport path is In Service, the data interface is available to be placed into service.

It should be noted that the data and/or voice services may be carried over ATM, data using AAL5, voice using AAL1 circuit emulation service (CES) (per ATM CES v2.0) with channel associated signaling (CAS) (per ANSI/TIA/EIA-464-B) for
10 signaling.

The following describes the implementation of the EOC, which conforms to TS 101 135 and T1E1.4/96-006, including framing, bit/byte assignments, addressing and a standard set of messages including those that allow the reading/writing of arbitrary, logical 'registers' in the HTU-R (HDSL Terminal Unit-Remote) by the HTU-C (HDSL
15 Terminal Unit-Central Office). Note that for this application, the standard EOC protocol is considered sufficiently reliable to obviate the need for higher-layer error detection/correction.

In the following paragraphs, the terms upstream and downstream are used. Downstream for a 2SDSL depends on whether it is an HTU-C or HTU-R. As an HTU-
20 C, downstream is toward the SDSL interface, and as an HTU-R, it is toward the backplane. Upstream is in the opposite direction. It is assumed that the IAD is always used in HTU-R mode, and thus, downstream is toward the customer (Ethernet) and upstream is toward the network (SDSL).

The EOC is used to command the far end to apply trunk conditioning (TC). In
25 order to put this in perspective, an overview of how TC, alarms and service states may be helpful. In Fig. 63, TC pertains to the data and/or signaling pattern used to overwrite a voice or data channel when an alarm condition is detected, the channel is put out of service, or TC is activated by the SM.

On an HTU-C RM 820A, the triggers for TC are as follows:
30

- Command from SM,
- loss of signal/loss of sync at SDSL line

-130-

- alarm indication from HTU-R (“local data interface OOS”)

On an HTU-R RM 820B, the TC triggers are

- Command from EOC
- loss of signal/loss of sync at SDSL line
- command from SM

5

If the HTU-R is an IAD 822, the TC triggers are

- Command from EOC
- loss of signal/loss of sync at SDSL line

The loss of signal/sync triggers affect all channels served by the HTU, while the other
10 triggers are specific to a particular cross connect/channel.

The HTU-C receives TC and Insertion Word (IW) values from the SM at the time of cross connect for use when TC is activated on that cross connect. Due to limitations of the IAD 822, the supplied TC values are ignored and a hard-coded TC takes place at the IAD, when TC is commanded to go active. The supplied TC values
15 are ignored and a hard-coded TC is applied in the upstream direction. The HTU-C performs no TC in the downstream direction. The HTU-C monitors for conditions that require activation of TC. The distinctions between the actions to take on a voice and data interface only apply to an HTU-C that is in IAD mode – in clear channel mode, the entire connection is treated as a data interface.

20 Upon detection of a TC trigger on a voice interface cross-connect, the HTU-C sends the TC Active command to the corresponding voice interface at the HTU-R via the EOC. When the TC trigger is no longer present, the HTU-C sends the TC Inactive command to the corresponding voice interface at the HTU-R via the EOC. If the HTU-C detects a TC trigger for a cross connect of the data interface, it sends the TC
25 Active command to the data interface at the HTU-R via the EOC. It also sends all ones toward the HTU-R across all affected DS0s.

The HTU-C monitors for conditions that require upstream TC. The normal event reporting to the SM continues to be performed. If a TC trigger is detected, the HTU-C sends all ones in the upstream direction on the affected cross-connects.

-131-

When TC on a given voice interface is active, ringing is halted and silence is transmitted on the line, as though the line was connected to a quiet termination. When TC on the data interface is active, any data received from the SDSL link is ignored and no data is transmitted downstream. If the HTU-R is an IAD, no Ethernet frames are
5 generated. If the HTU-R is a 2SDSL RM, all ones are transmitted in all affected timeslots.

When TC on a given voice interface is active, upstream TC at an IAD consists of sending the appropriate "On Hook" signaling bit pattern upstream, and silence in the channel. On an HTU-R RM, upstream TC consists of sending all ones upstream in the
10 affected DS0. If TC is active on a cross connect of the data interface. The HTU-R RM sends all ones upstream across all affected DS0s and the IAD sends no frames upstream. When TC is active on the data interface, the HTU-R sets the "Local Data Interface OOS" bit in the HTU-R Status register and clears it when TC is inactive.

Having described at a system level the functions of the 2SDSL RM 820, a
15 description of the module is now presented with reference to Fig. 64.

The module can be provisioned in software to operate at any of the following rates 256K, 384K, 512K, 768K, 1152K, 1536K, 1920K, 2048K, 2304K bits per second which results in the transmission of 4, 6, 8, 12, 18, 24, 30, 32 or 36 DS0 time slots.

Data across the DSL 821 (Fig. 60) is coupled from the line through Line
20 Interface 850. The Line Interface contains the hybrid, filters, HDSL transformer, lightning protection, sealing current and metallic test access.

From the line interface, the data passes through a transceiver 852 and channel unit 854 which are responsible for handling the data according to the HDSL standard. An FPGA 856A, 856B passes the data to a Universal Timeslot Interchanger (CT812)
25 858 and then to a bus switch 860 and from there to the backplane of the system. The message channel 30 is used for passing messages from/to the SM 21 through the bus switch, the CT812 and HDLC controller 862.

The transceiver (bit-pump) 852 has a built-in frequency synthesizer which allows variable rate configuration (144kbps to 2320kbps). The synthesizer uses a single
30 10.24 MHz crystal as reference. The receive section performs remote unit clock

-132-

recovery through an on-chip phase lock loop (PLL) circuit. The device also manages the startup and performance monitoring operations.

In the receive direction, data is received from the DSL link and passed to an internal digital signal processor (DSP) portion of the transceiver 852 which is responsible for extracting the far end 2B1Q encoded signal. The output of the transceiver is sent to the channel unit 854 on RDAT, BCLK and QCLK lines. RDAT outputs the serial quaternary data, BCLK is the bit-rate (two times symbol rate) clock output and QCLK runs at the symbol rate.

In the transmit direction, the digital symbols arrives from the channel unit 854 on the TDAT input and are transformed to an analog signal. The analog signal is conditioned to minimize crosstalk to adjacent subscriber lines. This conditioning enables the output signal to meet the standard requirements for pulse shape, power spectral density and output power at 784 kbps, 1168 kbps, and 2320 kbps without any changes required to external components, including the line transformer. The symbols are optionally scrambled. Isolated pulses can also be generated to support the testing of pulse templates.

The channel unit 854 is divided into two sections: PCM section 854A and HDSL section 854B. The PCM section is connected to the CT812 858 (through the FPGA 856A) and the HDSL section is connected to the transceiver 852.

The PCM section controls the flow of serial data between PCM and HDSL sections, establishes alignment between PCM and HDSL frames, and maintains synchronization between PCM and HDSL clocks. The PCM serial data is a framed data signal that consists of 64 8-bit timeslots. The PCM timebases create a 6 ms frame period based on the transmit clock (TCLK) and the receive clock (RCLK). The PCM timebase is programmed to equal approximately the HDSL 6 ms frame period in relation to the HDSL section's Bit Clock (BCLK_n). The resultant PCM and HDSL 6 ms frame intervals are used to establish alignment between PCM and HDSL frames, to maintain synchronization between transmit clocks by performing bit stuffing, and to recover PCM receive clock by comparing phase offset between frames.

The HDSL section is responsible for the following: sampling and aligning the incoming sign and magnitude data (see Table 40), scrambling/descrambling of payload

-133-

data, error performance monitoring, payload mapping of HDSL data from received HDSL frames into a PCM frame (and from transmit PCM frame into a HDSL frame), assembly of HDSL output frames and disassembly of HDSL receive frames.

- Each HDSL frame is nominally 6 ms in length and consists of 48 payload blocks
- 5 with each block containing a single Z-bit, plus a number of 36 payload bytes. Groups of 12 payload blocks are concatenated and separated by an ordered set of HDSL overhead bits, in which a 14-bit SYNC word pattern identifies the starting location of the HDSL frame. Fifty overhead bits are defined in one HDSL frame, but the last 4 STUFF (sq1-sq4) bits are nominally present in alternate frames. Therefore, one frame contains
- 10 an average of 48 overhead bits.

First Bit (Sign)	Second Bit (Magnitudes)	Quaternary Symbol (Quat)
1	0	+3
1	1	+1
0	1	-1
0	0	-3

TABLE 40 - 2B1Q Symbol Coding

- In the receive direction, the channel unit 854 gets a serial stream of data on the RDAT1 input which is sampled on the falling edge of BCLK1. The serially encoded 2B1Q sign bit is sampled when QCLK1 is low, and the 2B1Q magnitude bit is sampled
- 20 when QCLK1 is high.

The BCLK1 operates at twice the 2B1Q symbol rate (QCLK1 rate). The falling edge samples the QCLK1 and RDAT1 inputs. In the transmit direction, the rising edge of BCLK1 outputs the transmit data on TDAT1.

The FPGA 856A, 856B support the following (on both channels):

- 25 1. Passing data, clock and sync between the CT812 858 to the channel unit 854.

-134-

2. Converting the downstream signaling frame into an IAD signaling frame format (see Figs. 62A-62C).
3. Synchronizing the received data to the local clock and sync by using elastic storage.
- 5 4. Providing access to the FPGA's revision number and slot ID through blocks 880, 878.
5. Providing access to the upstream signaling data of 30 voice connections.
6. Controlling/monitoring the bus and ID switches 860, 861 through blocks 872, 874, 876.
- 10 7. Controlling 'sealing current' and 'metallic test access' relays through block 886.
8. Controlling face-plate LEDs through block 884.
9. Handling interrupts through blocks 890, 892.

Fig. 65 shows the data, clock and sync flow between the CT812 858, FPGA 15 856A, 856B and CU 854.

In the downstream direction, the FPGA gets the data and voice information for channel #1 from the CT812's SO_0 output and the signaling information from SO_4. The data and voice information for channel #2 is received from the CT812's SO_1 output and the signaling information from SO_5. The data rate of each of the SO 20 outputs is 4.098Mhz, i.e., 64 timeslots of DS0.

The channel units receive on the TSER lines a data stream of 4.098Mhz that carries the Data, Voice and Signaling information (when IAD mode is selected) as shown in Fig. 62A-62C.

When IAD without POTS mode is selected, i.e., Fig. 62B, the PCM frame does 25 not carry a Signaling timeslot and the Voice/Data timeslot starts at timeslot #2. When Clear Channel mode is selected, i.e., Fig. 62C, the frame does not carry either a Reserved nor Signaling timeslot and the Voice/Data timeslot will start at timeslot #1.

The FPGA gets the local clock and local frame sync and sends these signals to the channel units.

30 In the upstream direction, the FPGA gets the Data, Voice and Signaling from the channel units (in the same format as shown above) on RSER. The The FPGA also gets

-135-

the RCLK and RMSYNC signals from the CU. These signals are used by the elastic storage of the FPGA (described below). The FPGA send the received information from CU#1 to the CT812's SI_0, and the received information from CU#2 to the CT812's SI_1.

- 5 As described further herein, the FPGA provides a signaling frame format conversion referred to as a "swizzler" function. The swizzler block 857 converts the downstream signaling frame structure sent by the CT812 to a different signaling frame structure for use by the IAD (CPE). The difference between the two structures is the order in which the signaling bits are aligned in the frame. For instance, one could set
- 10 the following four Voice cross-connects for transport downstream to an IAD:
 - 1. DSX's timeslot #5 to 2SDSL's timeslot #1
 - 2. DSX's timeslot #21 to 2SDSL's timeslot #2
 - 3. DSX's timeslot #3 to 2SDSL's timeslot #3
 - 4. DSX's timeslot #14 to 2SDSL's timeslot #4
- 15 In this case, the swizzler function re-orders the signaling bits and the first signaling timeslot corresponding to signaling DS0 832 (Fig. 62A) appears as:

P1	P0	S5	S21	S3	S14	F	R1
----	----	----	-----	----	-----	---	----

- The data received from the channel units is passed through an elastic store to compensate for phase variations between the backplane bus and the SDSL. Data from
- 20 the HDSL to the system backplane is synchronized to the CT812 timing controls. Transmitted data over the DSL is passed through the FPGA without an elastic store.

- The upstream signaling portion 859 of the FPGA 856A is shown in Fig. 66. This block looks for a valid signaling timeslot in the upstream direction and then extract the appropriate A, B, C, D signaling bits of all the possible 30 far-end POTS
- 25 connections. The signaling information for the 30 POTS is carried in the second time-slot position of the HDSL upstream frame as noted previously. In order to get all the

-136-

signaling information of all the 30 POTS, the signaling block 859 processes the signaling timeslot 832 (Fig. 62A) of 24 consecutive DSL frames.

- Referring again to Fig. 66, the ‘Signaling T.S Extraction’ block 894 identifies the signaling timeslot 832 (Fig. 62A). The extracted timeslot is transferred to the
- 5 5 ‘Signaling Framer’ 899 that checks the phase (P1 and P2) bits sequences. This block will indicate a ‘valid Signaling T.S’ when its state machine identifies a valid phase bit sequence in six consecutive frames and indicate ‘Not-valid Signaling’ when one of the frame contains in valid phase bits.

The ‘Signaling Register Bank’ block 898 holds the current signaling bits of all

10 10 the 30 POTS per channel. This block sends a ‘Valid’ indication to the ‘CPU I/F’ block 896 when all the registers are loaded with valid information. There are 30 registers for both channels. Each register holds the A, B, C and D bits of two POTS lines. For example, register #1 holds the signaling bits of Ch1's POTS#1 (low 4 significant bits) and Ch1's POTS#2 (high 4 significant bits). Register #16 holds the signaling bits of

15 15 Ch2's POTS#1 (low 4 significant bits) and Ch2's POTS#2 (high 4 significant bits).

The CT812 (also called the ‘time slot interchange’) 858 can route any time slot location from/to the backplane side to/from any time slot location of the local side. There are 32 lines connected to the backplane. Each line is bi-directional and its data rate is 8.192Mhz (128 DS0s). The local side has 8 lines of transmit outputs and 8 lines

20 20 of receive inputs. Only 4 are being used in the transmit and the receive sides. Each line passes data at 4.098Mhz (64 timeslots).

The CT812 gets external clocks and sync from the backplane including the SCLKX2 (16.384MHz) SCLK (8.192MHz) and FSYNC. The device generates a local clock (L_CLK) and frame sync (L_FS) signals. The CT812 provides an interface

25 25 between the backplane Message Channel signal 30 and local HDLC controller 862.

The bus switch 860 is designed to power up in a high-impedance state. This isolates the module from the backplane’s TDM bussed signals, clocks and Message Channel and, therefore, provides a hot swap capability. The bus isolation module is controlled by switch 861 that in turn is under the control of the shelf manager SM 21.

30 The HDLC controller 862 handles the messages transferred between the on-board CPU and the shelf manager SM 21. An 80C51XA micro-controller 864 provides

appropriate timing, control and communication functions between the system and the board. The 80C51XA clock speed is 20.2752MHz.

The CPLD 866 performs address decoding, generates control signals and chip selects for the CPU peripherals, generates wait states for the CPU when it accesses

- 5 slower memory mapped devices and generates the clocks required by the CT812.

As described above, the signaling bit swizzler block 857 in the 2SDSL RM 820 (Fig. 64) is designed to receive downstream signaling data in the sequence it is presented and re-order the output sequence of signaling bits. The actual re-ordering is programmable via software and is dependent on the cross-connects to the SDSL RM.

- 10 The manner whereby the LAS 14 shelf transfers signaling information between cards via DS0s on the backplane is described hereinabove with reference to Fig. 16.

During each frame, a signal slot timeslot #2 (832 Fig. 62A) assigned to carry signaling information is received by the SDSL FPGA. The bit format of each signaling slot has been shown and described with reference to Table 8.

- 15 For the SDSL signaling bit swizzler 857, the Reserved bit (b8) is also treated as a valid signaling bit for use with E1 digital signal transmission format. As each frame is received, signaling bits are extracted from the correct bit position in each signaling slot to form the extended signaling channel matrix shown hereinabove at Table 9.

- 20 The operation of the signaling swizzler 857 is now described with reference to Fig. 67. The swizzler 857 includes an output sequencer register file bank 902, output sequencer phase generator 904 and input sequencer phase detector 906. Also included are 30:1 multiplexer 910, index counter 912, address counter 908, dual port memory 914, 60:1 multiplexer 916 and 4:1 multiplexer 918.

- 25 As the Extended Frame format suggests, each phase of the ESF consists of six frames of 5 signaling bits each. Thus, for each phase, a thirty bit memory 914 is used to store the signaling information. This memory has the following format shown in Table 41.

-138-

SIGNALING MEMORY		
	Address	Signal Bit
5	0x00	S ₁
	0x01	S ₂
	0x02	S ₃
	0x03	S ₄
	0x04	R ₁
	0x05	S ₅
	0x06	S ₆
10	0x07	S ₇
	0x08	S ₈
	0x09	R ₂
	0x0A	S ₉
	0x0B	S ₁₀
15	0x0C	S ₁₁
	0x0D	S ₁₂
	0x0E	R ₃
	0x0F	S ₁₃
	0x10	S ₁₄
20	0x11	S ₁₅
	0x12	S ₁₆
	0x13	R ₄
	0x14	S ₁₇
	0x15	S ₁₈
25	0x16	S ₁₉
	0x17	S ₂₀
	0x18	R ₅
	0x19	S ₂₁
	0x1A	S ₂₂
30	0x1B	S ₂₃
	0x1C	S ₂₄

-139-

0x1D	R ₆
0x1E	0
0x1F	1

TABLE 41.

5 To control the output sequence, the signaling bit swizzler 857 provides thirty, 5-bit registers in file 902 which are cycled sequentially by the read (or output) sequencer 904 using index counter 912 and 30:1 mux 910. Each of these registers holds the 5-bit address of the desired signaling bit to be output in the desired order. Table 42 describes the 30-output Register Bank.

10

OUTPUT REGISTER BANK		
Address	Register	Output Signal Bit Location
0xDE0XX	R ₀	1
0xDE1XX	R ₁	2
0xDE2XX	R ₂	3
0xDE3XX	R ₃	4
0xDE4XX	R ₄	5
0xDE5XX	R ₅	6
0xDE6XX	R ₆	7
0xDE7XX	R ₇	8
0xDE8XX	R ₈	9
0xDE9XX	R ₉	10
0xDEAXX	R _A	11
0xDEBXX	R _B	12
0DECXX	R _C	13
0DEDXX	R _D	14
0DEEXX	R _E	15
0DEFXX	R _F	16
0DF0XX	R ₁₀	17
0DF1XX	R ₁₁	18
0DF2XX	R ₁₂	19
0DF3XX	R ₁₃	20
0DF4XX	R ₁₄	21
0DF5XX	R ₁₅	22
0DF6XX	R ₁₆	23
0DF7XX	R ₁₇	24
0DF8XX	R ₁₈	25
0DF9XX	R ₁₉	26
0DFAXX	R _{1A}	27
0DFBXX	R _{1B}	28
0DFCXX	R _{1C}	29
0DFDXXX	R _{1D}	30

TABLE 42

-140-

- The contents of this register bank 902 in addition to the Signaling Memory 914 determine which of the thirty signal input bits are output and in what sequence. For example, to connect signaling bit S_{17} to output signaling bit 1, the address of S_{17} in the Signaling Memory (0x14) is loaded into R_0 . To connect the input signaling bit S_{21} to 5 output signaling bit 2, the address of S_{21} in Signaling Memory (0x19) is loaded to R_1 . Only those signaling bits which are to be used need to be programmed into the register bank. However, by loading the unused registers with 0x1F, all remaining signaling bits are set to '1. If the registers are loaded with 0x1E, all remaining signaling bits are set to '0.
- 10 The actual output format of the signaling data is illustrated in Table 43.

SIGNALING BIT SWIZZLER OUTPUT FORMAT								
FRAME #	P1	P0	S1	S2	S3	S4	F	R
1	0	0	$S(R_0)$	$S(R_1)$	$S(R_2)$	$S(R_3)$	1	$R(R_4)$
2	0	0	$S(R_5)$	$S(R_6)$	$S(R_7)$	$S(R_8)$	1	$S(R_9)$
3	0	0	$S(R_A)$	$S(R_B)$	$S(R_C)$	$S(R_D)$	1	$S(R_E)$
4	0	0	$S(R_F)$	$S(R_{10})$	$S(R_{11})$	$S(R_{12})$	1	$S(R_{13})$
5	0	0	$S(R_{14})$	$S(R_{15})$	$S(R_{16})$	$S(R_{17})$	1	$S(R_{18})$
6	0	0	$S(R_{19})$	$S(R_{1A})$	$S(R_{1B})$	$S(R_{1C})$	1	$S(R_{1D})$

TABLE 43

Where $S(R_x)$ implies the signaling bit in the Signaling Memory indirectly indexed by the contents of Register (x).

- 15 The following example illustrates the swizzler behavior. Assume that the register bank 902 is loaded with the following values.

-141-

OUTPUT REGISTER BANK		
Address	Register	Data Value
0xDE0XX	R ₀	0x07
0xDE1XX	R ₁	0x0A
0xDE2XX	R ₂	0x0E
0xDE3XX	R ₃	0x12
0xDE4XX	R ₄	0x04
0xDE5XX	R ₅	0x1C
0xDE6XX	R ₆	0x19
0xDE7XX	R ₇	0X00
0xDE8XX	R ₈	0X02
0xDE9XX	R ₉	0X09
0xDEAXX	R _A	0X05
0xDEBXX	R _B	0X16
0xDECXX	R _C	0X01
0xDEDXX	R _D	0X08
0xDEEXX	R _E	0X0D
0xDEFXX	R _F	0X1F
0xDF0XX	R ₁₀	0X1F
0xDF1XX	R ₁₁	0X1F
0xDF2XX	R ₁₂	0X1F
0xDF3XX	R ₁₃	0X1F
0xDF4XX	R ₁₄	0X1F
0xDF5XX	R ₁₅	0X1F
0xDF6XX	R ₁₆	0X1F
0xDF7XX	R ₁₇	0X1F
0xDF8XX	R ₁₈	0X1F
0xDF9XX	R ₁₉	0X1F
0xDFAXX	R _{1A}	0X1F
0xDFBXX	R _{1B}	0X1F
0xDFCXX	R _{1C}	0X1F
0xDFDXX	R _{1D}	0X1F

TABLE 44

-142-

For this example, the output signaling slot data is that shown in Table 45:

SIGNALING BIT SWIZZLER								
FRAME #	P1	P0	S1	S2	S3	S4	F	R
1	0	0	A ₇	A ₉	A ₁₃	A ₁₆	1	AR ₁
2	0	0	A ₂₄	A ₂₁	A ₁	A ₃	1	AR ₂
3	0	0	A ₅	A ₁₉	A ₂	A ₉	1	AR ₃
4	0	0	1	1	1	1	1	1
5	0	0	1	1	1	1	1	1
6	0	0	1	1	1	1	1	1
7	0	1	B ₇	B ₉	B ₁₃	B ₁₆	1	BR ₁
8	0	1	B ₂₄	B ₂₁	B ₁	B ₃	1	BR ₂
9	0	1	B ₅	B ₁₉	B ₂	B ₉	1	BR ₃
10	0	1	1	1	1	1	1	1
11	0	1	1	1	1	1	1	1
12	0	1	1	1	1	1	1	1
13	1	0	C ₇	C ₉	C ₁₃	C ₁₆	1	CR ₁
14	1	0	C ₂₄	C ₂₁	C ₁	C ₃	1	CR ₂
15	1	0	C ₅	C ₁₉	C ₂	C ₉	1	CR ₃
16	1	0	1	1	1	1	1	1
17	1	0	1	1	1	1	1	1
18	1	0	1	1	1	1	1	1
19	1	1	D ₇	D ₉	D ₁₃	D ₁₆	1	DR ₁
20	1	1	D ₂₄	D ₂₁	D ₁	D ₃	1	DR ₂
21	1	1	D ₅	D ₁₉	D ₂	D ₉	1	DR ₃
22	1	1	1	1	1	1	1	1
23	1	1	1	1	1	1	1	1
24	1	1	1	1	1	1	1	1

TABLE 45

The output signaling byte with the swizzled signaling data is inserted into slot 2 of the FPGA output frame as indicated previously with respect to Fig. 62A. The swizzler

- 5 inserts all ones into slot 2 whenever the input sequencer 906 is out of, or seeking synchronization with the incoming P bits. If any error occurs during the reception of the signaling, the sequencer 906 will reset and try to re-synchronize to the incoming P bits. Again, all ones are output until the proper Pbit synchronization is achieved.

SMU2

- 10 The SMU2 described above with reference to Fig. 7 is a line card capable of system clock generation, user interface, and alarm control. An embodiment of a shelf manager SMU2 with an improved processing core, added memory, Ethernet interface,

-143-

PCMCIA FLASH disk interface, and a SCSA bus interface device to allow the SM direct access to backplane DS0s is now described with reference to Fig. 68.

- The SMU2 includes MPC860MH processor 930, FLASH program memory 932, DRAM memory 934, Non-Volatile SRAM memory 936, an optional on-board
- 5 PCMCIA FlashDisk 938, a CT-812 SCSA Universal Timeslot Interchanger 940, a Field Programmable Gate Array (FPGA) 942, Real Time Clock 944, on-board power supplies 946, 948, power/battery monitor 950, SCSA Bus isolation switches 951, 953, Ethernet support circuitry 954, and various clock oscillators and VCXO 956 for system timing.

- The SMU2 utilizes the Motorola MPC860MH as the primary processor 930.
- 10 This processor runs at a clock speed of 25MHz that is derived from an external 32.768KHz crystal and an internal PLL. The MPC860 has a 32-bit data bus as well as a 32-bit address bus and incorporates a superscaler internal architecture with both data and instruction caches. This architecture results in enhanced performance with execution rates greater than 1 instruction/clock cycle.
- 15 The MPC860 contains a fully configurable, 8-bank memory controller allowing for glueless interface to FLASH, DRAM, NVSRAM, and the other peripheral devices in the SMU2. This memory controller utilizes a General Purpose Chip-select Machine (GPCM) and a pair of User Programmable Machines (UPM) for complex sequencing and timing. The memory controller also provides the capability for burst operations,
- 20 wait state insertion, internal/external transfer acknowledgement, and memory range monitoring.

- The SMU2 also utilizes an integrated PCMCIA host adapter/controller 960 of the MPC860 for access to the on-board, fully captive, solid state FlashDisk 938. This controller requires only external drivers and transceivers to implement a fully compliant
- 25 PCMCIA 2.1 True IDE interface.

- The SMU2 also uses the digital I/O ports 962 of the MPC860. These ports are used for generating signal timing/sequencing for FPGA download and accessing the serial real time clock (RTC) 944.

- A prominent feature of the MPC860 utilized by the SMU2 is the
- 30 Communications Processor Module (CPM). The CPM provides a flexible and integrated approach to the SMU2's communications-intense environment. The CPM has its own

-144-

RISC communications processor optimized for serial communications. This RISC processor can service several integrated communications channels, performing low-level protocol processing and controlling serial channel DMA operations. The MPC860MH provides two full duplex serial management controllers (SMC) and four 5 full-duplex serial communications channels (SCC). SMC1 964 is used by the SMU2 to provide an RS-232, text-based interface to the system user via both the front panel of the SMU2 and also via a redundant RJ-45 connector 965 on the backplane. This interface operates as a subset of the system management software.

The SMU2 uses SCC1 966 of the MPC860 to provide a 10BASE-T Ethernet 10 interface to the backplane 941 for support of IEEE 802.3 CSMA/CD LAN protocol. This interface is also used as a high-speed alternative to the SMC1 interface. The external PHY level Ethernet transceiver and associated transformer 954 are the only external components necessary to implement this protocol compliant interface.

SCC2 968 of the MPC860 is used by the SMU2 in HDLC bus mode to provide 15 communications over the system Message Channel. The SMU2 operates this channel at 2.048Mb/s and uses a 32 data-byte frame format as described hereinabove. The message channel is implemented as a single line in bus mode that allows for point to multipoint LAN operations. The MPC860 is configured in this mode to utilize the option for hardware collision detection and retransmission.

20 SCC4 970 of the MPC860 is also used by the SMU2 in HDLC mode but interfaced to the CT-812 TSI 940 via a Time Division Multiplexed (TDM) serial controller. Using the internal Time Slot Assignor (TSA) and framing control strobes from the CT-812, the normally contiguous HDLC stream is broken up in 8 bit slices and injected into one of 32 slots within a frame. Although each individual message takes 25 longer to transmit, this feature allows for multiple simultaneous messaging which can utilize the full bandwidth of the 2048-timeslot backplane. This feature is useful for communication with other Resource Modules.

The SMU2 utilizes an arsenal of memory devices for various purposes. These include PCMCIA FlashDisk 938s on-board FLASH EPROM 932, Fast Page Mode 30 (FPM) DRAM 934, and battery backed SRAM 936.

-145-

The SMU2 incorporates two 1M x 16 bit FLASH EPROM devices 932 for boot code storage. These devices are accessed via chip select 1 from the MPC860 using the GPM in the MPC860's memory controller. Using 90ns devices, instructions can be read/written using only a single wait state @25MHz processor clock speed.

- 5 The SMU2 incorporates two 4M x 16 bit FPM DRAM devices 934 for temporary storage requirements. This results in a total DRAM capacity of 16MBytes. Access to these devices is via chip select 1 of the MPC860 using the UPM for signal/strobe timing and sequencing. Although configured as 32 bits, read/write accesses to this DRAM bank can be as 8-bit byte, 16-bit word, or 32-bit long word.
- 10 Using the 60ns devices specified, single beat accesses can be achieved using 1-wait states and a total access time of three clock cycles. However, because these devices are fast page mode, multi-word burst accesses are supported which greatly reduces the average access time. These burst accesses are essential for cache fill operations that ultimately enhance the MPC860's performance.
- 15 The SMU2 has a single 128K x 8 bit Static RAM 936 that is backed-up by the on-board lithium battery 952 when power is removed from the board. Access to this device is via chip select 4 of the MPC860 using the GPM to control signal strobes and timing. This 85nS device requires only a single wait state for normal operation.

The SMU2 has an on-board PCMCIA interface (socket) configured for True-IDE mode operation. Connected to this socket is a Type II PCMCIA solid state FlashDisk 938 which can provide storage with capacities up to 300MBytes. This device is used by the MPC860 like a hard disk and stores program information that is accessed after the SMU2 boots from FLASH. Access to this device is via the PCMCIA adapter of the MPC860 with the addition of external drivers and transceivers.
- 25 The SMU2's requirement for full access to the Signal Computing System Architecture (SCSA) backplane bus 941 is accomplished through the use of the CT-812 Universal Time Slot Interchanger 940. This device provides a connection between telephony or signal processing resources and the backplane bus. This device is accessed by the MPC860 for configuration and status through an 8-bit port using chip select 2 and the MPC860's GPM memory controller. Actual CT-bus TDM data is interfaced through the CT-812 to the MPC860 via SCC3 in TDM mode using the Time Slot

-146-

Assignor. Timing for the CT-812 is provided by the on-board 16.384MHz Stratum IV oscillator with "Frame Sync" generated by the SMU2's field programmable gate array (FPGA). Although the CT-812 has the capability of supporting 32 CT bus lines, only 16 are used in the backplane.

- 5 The SMU2 uses a battery backed Real Time Clock module 944 with integrated crystal to provide the system shelf with a true clock resource. This device is accessed via digital I/O port pins on the MPC860. Accessed as a 52-bit serial frame, the MPC860 provides the data, clock and write control strobes to read/write this device. Using the on-board lithium battery, this device stays operational in low power mode
- 10 when power is removed from the SMU2. Accuracy of the device is within two minutes per month.

The SMU2 has six alarms relays 972 connected to twelve signal lines that connect to the backplane and are accessible via terminal blocks on the backplane. The alarms are broken into two categories, visual and audible and are further defined by

15 three levels of severity: minor, major and critical. The alarm relays are used by the system shelf to indicate fault conditions to the outside world. Each relay has an associated jumper to configure the alarm contact for either normally open or normally closed. Since the Relays are sourced by the -48V supply, the SMU2 also incorporates two TTL level converters to enable the relays. The relays are under control of the

20 processor through a register within the FPGA.

The FPGA 942 of the SMU2 contains the logic necessary to interface to the rest of the system. The FPGA contains the logic to perform tasks including clock recovery, clock generation, status I/O, phase lock loop (PLL), alarm control, and automatic one-wire interface.

- 25 The FPGA contains logic that is used in conjunction with an external low pass filter 957 and voltage controlled crystal oscillator (VCXO) 956 to form a phase lock loop (PLL) used for clock recovery. The VCXO operates with a center frequency of 32.768 MHz. The frequency is adjusted depending upon the pulse width output by the phase discriminator within the FPGA. These output pulse streams are integrated via the
- 30 LPF and result in a bias voltage which controls the frequency. The reference source for the PLL is selectable via software from one of 5 sources: the on-board 16.384MHz

Stratum IV oscillator, the central office composite clock, or loop recovered clocks 1, 2, and 3. The loop-recovered clocks are derived from input data streams in either the BRI or DSX resource modules. The FPGA also contains logic that monitors the integrity of each selected clock and reports errors to the processor via interrupt.

5 The FPGA contains the logic for clock generation based upon the output of the 32.768MHz VCXO. Through the use of counters, the FPGA generates SCLKx2 and FSYNC signals which are connected to each RM in the backplane and synchronize all operations.

10 The FPGA also contains logic used to provide status and other information from the backplane to the processor. Acting as a buffer, the FPGA provides the processor access to the 5-bit backplane revision field, the 5-bit slot ID field, the 4-bit shelf ID field, and the 3-bit alarm input field. The FPGA also provides access to backplane status signals CLKFAIL and CTERM.

15 Additionally, the FPGA contains status output bits to control other devices on the SMU2 circuit board. These devices include the three front panel LEDs, the alarm relays, and the enables for the isolation bus switches, and the system CLKFAIL signal.

20 The FPGA of the SMU2 also contains the logic to implement an automatic one-wire protocol sequencer. Each RM in the system shelf has an individual line to the SMU2 that is used to enable or disable the RM. The addressable switch on each RM requires a 72-bit data stream to turn on or turn its bus switches. The timing of this 72-bit stream is critical and is controlled by the automatic one-wire sequencer within the FPGA. The processor starts a sequence by setting a single bit and is interrupted via the FPGA 2ms later when the automatic sequence completes. Errors during the sequence result in interrupts to the MPC860. The processor controls the selection of 1 of 25 one-wire interfaces on the FPGA. More information about the one-wire protocol is available in the 1405 Addressable Switch product descriptin from Dallas Semiconductor.

25 The FPGA is accessed by the processor as an 8-bit device using chip select 3 and requires no wait states for normal operation.

30 Each resource module and the SMU2 can be fully isolated from the SCSA backplane through the use of bus switches 951, 953. These switches are disabled unit enabled via the processor on each RM. They may be disabled by the SMU2 if the shelf

-148-

manager determines that the offending RM is causing system failure. The signals that are isolated on the SMU2 via bus switches are the 25 one-wire control signals, system clocks, system frame sync, the 16 CT bus lines, and the message channel line. On the SMU2, the bus switches are controlled via two signals, BUSEN and CLKEN that are

- 5 controlled by the processor through the FPGA.

EPRM2

The following describes an Ethernet protocol resource module (EPRM2Z). The EPRM2 Module is a protocol adapter resource module which concentrates encapsulated MAC frames from multiple subscribers into a single 10/100Base-T

- 10 interface.

The EPRM2 provides the following features:

- 4-wire 10Base-T/100Base-TX interface at the backplane
- Support for xDSL subscribers at Nx64 Kbps, N=2-32
- Maximum bandwidth equivalent to 256 full duplex DS0 timeslots or
15 16.384 Mbps
- Support for MAC layer bridging
- Interoperability with Integrated Access Devices
- Interoperability with iDSL Network Extenders

The EPRM2 allows Internet Service Providers (ISP) to connect to their
20 customers without having to use traditional switching facilities in central offices. Fig. 69 shows this configuration. The EPRM2 1200 provides concentration and 10/100Base-T backhaul for small businesses and SOHOs at iDSL device 1205 via iDSL 1201 and larger businesses and “extranets” at IAD 1207 via SDSL 1203. In this arrangement, the subscribers are connected using BRI and SDSL circuits. In particular,
25 the iDSL subscribers are connected to the shelf 14 through multiple 4BRI circuit packs (not shown).

The EPRM2 1200 is essentially a bridge between 10/100Base-T and Ethernet frames encapsulated in the backplane DS0s, with the Ethernet port optionally acting as an “uplink” port. The EPRM2 1200 conforms to those requirements specified in IEEE
30 Std 802.1Q-1998.

-149-

The EPRM2 relays individual MAC user data frames between its various ports. A port may be either an Ethernet port or an HDLC port, which is dynamically configured by the management system via a cross-connect to the backplane. The HDLC ports are ultimately connected to a similar device (e.g., an IAD), which performs a
5 translation back to Ethernet.

The EPRM2 1200 includes an Ethernet port that sends and receives Ethernet Frames in accordance with IEEE 802.3. The receive errors that are detected are those defined in IEEE 802.3, namely, a) frame length inconsistent with the length value specified in the Ethernet Frame length/type field, (when it contains a length); b) not an
10 integral number of octets; or c) the generated CRC does not match the one received. Additionally, “runt” frames, (as defined in IEEE 802.3 paragraphs 4.4.2.1 and 3.1.1) are considered a receive error. Frames received with errors are discarded.

The Ethernet port supports 10BASE-T and 100BASE-TX, and the speed is a selectable option or may be allowed to set itself through “auto-negotiation”. The
15 Ethernet port supports Half and Full Duplex mode.

The EPRM2 1200 includes one or more HDLC ports which send and receive Ethernet frames encapsulated in HDLC per RFC 1662 using a selectable format. The supported selections include:

- Compressed HDLC (C-HDLC) (as described in “Address and Control
20 Field Compression” in paragraph 3.2 of RFC 1662)
- Standard HDLC (HDLC)
- PPP

Fig. 70 shows these WAN encapsulation details with respect to the Ethernet frame.

25 The following describes an EPRM2 1200 embodiment as shown in Fig. 71. A Motorola MPC860 serves as the processing platform 1202 for the EPRM2. This device utilizes both a 32-bit data and address bus. The functionality of this processor has been described above with respect to the SMU2 and Fig. 68.

The EPRM2 supports 16MB of DRAM memory 1204 consisting of two, 8MBit
30 parts configured as 4M x 32bits. This memory can be used for both program or data

-150-

storage. This memory interfaces directly to the MPC860 processor via the UPM memory controller port.

The EPRM2 supports 512KB, 1MB, and 4MB of flash memory 1206 organized as two, 16-bit devices in a 32-bit data bus configuration. This memory can be used for
5 program storage only. This memory interfaces directly to the MPC860 processor via the GPM memory controller port.

The EPRM2 incorporates an FPGA 1208 which contains clock generation and frame synchronization circuitry as well as various system control registers. These system control registers are used for LED control, clock control and input status. The
10 FPGA is accessed as a byte peripheral device and is selected via a MPC860 chip select.

The EPRM2 has access to the system SCSA bus 941 via a CT-812 Bus System Interface controller 1210. This access is provided both as parallel access and direct serial access to the backplane via one of the MPC860's HDLC channels configured for TDMA operation. This device is access as a byte-wide peripheral device and is selected
15 via a MPC860 programmable chip select.

Bus isolation and hot swap support is provided by a set of bus isolation FET switches. The CPU of the EPRM2 is capable of enabling or disabling the bus connection to the backplane or the clock connection to the bus via separate signals from the FPGA. The bus switches are designed to be inherently high impedance when the
20 circuit card has no power or is actively disabled.

The EPRM2 1200 further includes a multichannel synchronous communications controller 1212 (implemented as Conexant device MUSYCC-CN8478) and an Ethernet controller 1214. The Ethernet controller 1214 can be an Intel 82559 device which provides a Media Access Controller (MAC) 1218 and physical layer (PHY) interface
25 1216. In operation, multichannel HDLC encapsulated data from the backplane bus 941 is passed to the MUSYCC 1212 and transferred into RAM 1204. From RAM 1204, the data is then multiplexed into a single Ethernet channel via the Ethernet controller 1214. This process is bi-directional.

While this invention has been particularly shown and described with references
30 to preferred embodiments thereof, it will be understood by those skilled in the art that various changes in form and details may be made therein without departing from the

-151-

spirit and scope of the invention as defined by the appended claims. It will be apparent to those of ordinary skill in the art that methods involved in the present invention may be embodied in a computer program product that includes a computer usable medium. For example, such a computer usable medium can include a readable memory device, 5 such as a hard drive device, a CD-ROM, a DVD-ROM, or a computer diskette, having computer readable program code segments stored thereon. The computer readable medium can also include a communications or transmission medium, such as a bus or a communications link, either optical, wired, or wireless, having program code segments carried thereon as digital or analog data signals.

10 Those skilled in the art will know, or be able to ascertain using no more than routine experimentation, many equivalents to the specific embodiments of the invention described herein. These and all other equivalents are intended to be encompassed by the following claims.

-152-

CLAIMS

What is claimed is:

- 5 1. A system comprising:
 - a plurality of interchangeable resource modules for providing interfaces for a variety of network communications services;
 - a management unit for monitoring each resource module over a message channel; and
 - wherein each resource module communicates with the management unit over said message channel.
- 10 2. The system of claim 1, wherein a cross-connection is established by an operator using the interface between any resource module and any other resource module via messages from the management unit to the message channel and wherein the management unit maintains in memory, common settings and configurations for the resource modules and initializes and synchronizes timing functions in each resource module and provides a management interface for each module.
- 15 3. The system of claim 2 wherein the interface is accessible via the management unit and allows a user to manage system resources by connecting the unit to a menu based control screen to configure and perform maintenance on system components.
- 20 4. The system of claim 1 including a contention detection protocol wherein contention is resolved by having the module detecting contention immediately aborting its message and entering a high impedance state and remaining in that state until an idle condition is detected by the module in contention at which time the module may transmit a single frame of bits, but does not transmit
- 25

-153-

another frame of bits until at least 5 or more successive "one" bits are detected on the control channel.

5. 5. A communication system for providing subscribers access to a variety of networks services and protocols including ATM, frame relay and IP comprising:

10 a plurality of resource modules which provide an interface between the subscribers and the network services;

a management unit which monitors each resource module over a data bus, and wherein each management unit includes at least one processor which stores provisioning information for a respective resource module; and a processor in at least one resource module for receiving provisioning information from the management unit;

15 a message channel for providing communication between the resource modules and the management unit; and

a graphical user interface accessible via a faceplate on the management unit, the interface allowing a user to manage resources.

- 20 6. The system of claim 5 wherein the management unit downloads the provisioning information to a respective resource module over the message channel.

- 25 7. The system of claim 6 wherein each module is assigned one or more unique time slots of a data bus for reception and transmission of digital information and wherein a cross-connection is established by the management unit between any resource module and any other resource module, thereby providing full time slot interchangeability for each module.

8. The system of claim 7 including a bus contention protocol.

- 30 9. A graphical user interface for an access system which interface enables a user to exercise, configure and test units of the system, the interface comprising:
a display providing N possible function views including:

-154-

operation view, a cross-connect map view and an administration map view with each such view supporting N-entity levels wherein N = 3.

10. 5 The interface of claim 9 wherein the entity levels are shelf, equipment, and interfaces.
11. 10 A graphical user interface for an access system which system includes a shelf containing a plurality of modules arranged in side-by-side locations adjacent each other, each module having a separate face plate; the interface comprising:
15 a display partitioned into a plurality of views; one of which is a hierachal view which depends upon a users selection and wherein the first such view in the hierarchy represents the shelf in its entirety as then currently provisioned.
12. 20 The GUI of claim 11 including additional selectable views of individual modules, interfaces on the modules and cross-connections between interfaces.
13. 25 The GUI of claim 11 including a zone with views of several selectable indicia a first of which enables a user to proceed to another view in another zone or to revert to a previous view in the other zone; and a second of which enables a user to display in another zone processing, status/alarm reporting, and maintenance information for the various modules.
14. 30 The GUI of claim 13 further including third indicia which enable a user to establish crossconnections on the modules and a fourth of which enables the user to access and select administrative functions for the modules.
15. A communication system for providing subscribers access to a variety of networks services comprising:
30 a plurality of resource modules which provide an interface to the network services;

-155-

a management unit which monitors each resource module over a first bus; and wherein each resource module includes at least one processor which stores provisioning information for each module; and

5 a second bus for communicating between the management unit and the resource modules during start-up and wherein the first bus provides each module with the ability to read or write from any timeslot thereby achieving full time slot interchange capability; and

10 a graphical interface which enables a user to perform the following functions:

- (a) configure each resource module
- (b) cross-connect mapping resource modules
- (c) track the status of resource modules
- (d) report events, alarms and statistics
- (e) test and maintain resource modules.

15 16. The system of claim 15 wherein each resource module is connected to the management unit and access to, and software upgrade provisioning of the resource modules is provided over a message channel using a predefined set of protocol messages.

20 17. A communication system for providing subscribers access to a variety of networks services comprising:

25 a plurality of resource modules which provide an interface to the network services, each of which includes a transient protection circuit protecting its module from high-voltage transients and a rate conversion unit which converts the operating rate of its module to match a data rate of input data coupled to the module from a backplane input/output interface between the subscriber and the network services;

30 a management unit which monitors each resource module over a bus; and wherein each resource module includes at least one processor which stores provisioning information for each module; and

-156-

a graphical interface which includes a computer coupled to the system and enabling a user to perform several functions including operation, maintenance, and monitoring of the resource modules.

- 5 18. The system of claim 17 wherein each module is arranged in slots and can be interchanged, which are configured by the user from the interface via a message channel communicated over a bus.
- 10 19. The system of claim 18 wherein the slots are provided on a single shelf and the management unit is contained in one of the slots.
- 15 20. A method of interfacing a variety of network services with subscribers using a variety of resource modules having interfaces for establishing a data point between any two modules, cross-connecting one module to another and which modules are provisioned and maintained from a management unit operable from a user interface the management unit communicating of digital messages over a message channel, comprising the steps of:
- 20 a) providing transmit and receive time slot assignments from the management unit to the resource modules;
- b) establishing configurable parameters to define a valid cross-connection.
- 25 21. The method of claim 20 wherein the parameters include one or more of the following:
- a) identifying one interface on a first side of the cross-connection;
- b) identifying one interface on a second side of the cross-connection;
- c) providing the number of signaling bits required to support the cross-connection;
- d) providing an insertion word for transmission in the event an alarm condition occurs;

30

-157-

- 5 e) providing a value for the signaling bits to be transmitted from the interface of the first side of the cross-connection toward the second side of the cross-connection in the event an alarm condition occurs;
- f) providing a value for the signaling bits to be transmitted from the interface of the second side of the cross-connection toward the second side of the cross-connection in the event an alarm condition occurs; and setting a primary service state for the connection.

- 10 22. In a communication system having plural resource modules for providing communications services and a backplane communication bus for interconnecting the resource modules, wherein the bus has plural backplane timeslots including voice timeslots for carrying voice information and signaling timeslots for carrying signaling information for one or more corresponding voice timeslots, an improved resource module comprising:
- 15 a backplane interface circuit for selecting backplane voice and signaling timeslots from the backplane communication bus; and
- 20 a multiplexer circuit for multiplexing the selected backplane voice and signaling timeslots into a downstream signal, the multiplexer circuit including a signaling mapper for mapping the signaling information from each of the selected backplane signaling timeslots into a common downstream signaling timeslot.
- 25 23. The improved resource module of Claim 22 wherein the downstream signal has a frame format comprising M downstream timeslots including N<M downstream voice timeslots and the common downstream signaling timeslot.
- 30 24. The improved resource module of Claim 23 wherein the backplane communication bus further includes backplane data timeslots for carrying data information and wherein the downstream frame format further includes downstream data timeslots.

-158-

25. The improved resource module of Claim 22 further comprising a line interface circuit for converting the downstream signal to a digital subscriber line format for transmission to a downstream integrated access device.
- 5 26. The improved resource module of Claim 21 wherein the signaling information includes in-band signaling.
27. The improved resource module of Claim 26 wherein the signaling information includes ABCD signaling bits.
- 10 28. A communication system comprising:
 - plural resource modules for providing communications services;
 - a backplane communication bus for interconnecting the resource modules, wherein the bus has plural backplane timeslots including voice timeslots for carrying voice information and signaling timeslots for carrying signaling information for one or more corresponding voice timeslots,
 - 15 wherein at least one of the resource module comprises:
 - a backplane interface circuit for selecting backplane voice and signaling timeslots from the backplane communication bus;
 - 20 a multiplexer circuit for multiplexing the selected backplane voice and signaling timeslots into a downstream signal, the multiplexer circuit including a signaling mapper for mapping the signaling information from each of the selected backplane signaling timeslots into a common downstream signaling timeslot; and
 - 25 a line interface circuit for converting the downstream signal to a digital subscriber line format for transmission to a downstream integrated access device.

-159-

29. The communication system of Claim 28 wherein the downstream signal has a frame format comprising M downstream timeslots including N<M downstream voice timeslots and the common downstream signaling timeslot.
- 5 30. The communication system of Claim 29 wherein the signaling information includes in-band signaling.
31. The communication system of Claim 30 wherein the signaling information includes ABCD signaling bits.
- 10 32. In a communication system having plural resource modules for providing communications services and a backplane communication bus for interconnecting the resource modules, wherein the bus has plural backplane timeslots including voice timeslots for carrying voice information and signaling timeslots for carrying signaling information for one or more corresponding voice timeslots, a method of communication comprising:
 - selecting backplane voice and signaling timeslots from the backplane communication bus; and
 - multiplexing the selected backplane voice and signaling timeslots into a downstream signal, including mapping the signaling information from each of the selected backplane signaling timeslots into a common downstream signaling timeslot.
- 15 33. The method of Claim 32 wherein the downstream signal has a frame format comprising M downstream timeslots including N<M downstream voice timeslots and the common downstream signaling timeslot.
- 20 34. The method of Claim 33 further comprising converting the downstream signal to a digital subscriber line format for transmission to a downstream integrated access device.
- 25
- 30

-160-

35. In a communication system having plural resource modules for providing communications services and a backplane communication bus for interconnecting the resource modules, wherein the bus has plural backplane timeslots including voice timeslots for carrying voice information and signaling timeslots for carrying signaling information for one or more corresponding voice timeslots, an improved resource module comprising:
- 5 a backplane interface circuit for selecting first voice and signaling timeslots from the backplane communication bus and for putting second voice and signaling timeslots onto the backplane communication bus;
- 10 a line interface circuit for converting a downstream PCM signal to a downstream DSL signal for transmission to an integrated access device and for converting an upstream DSL signal from the integrated access device to an upstream PCM signal;
- 15 a multiplexer circuit for multiplexing the first voice and signaling timeslots into the downstream PCM signal; and
- a demultiplexer circuit for demultiplexing the upstream PCM signal to the second voice and signaling timeslots.
36. The improved resource module of Claim 35 wherein the downstream PCM signal has a frame format comprising M downstream timeslots including N<M downstream voice timeslots and a common downstream signaling timeslot.
- 20
37. The improved resource module of Claim 36 wherein the multiplexer circuit includes a downstream signaling mapper for mapping the signaling information from each of the first signaling timeslots into the common downstream signaling timeslot.
- 25
38. The improved resource module of Claim 37 wherein the signaling information includes ABCD type in-band signaling.
- 30

-161-

39. The improved resource module of Claim 35 wherein the upstream PCM signal has a frame format comprising M upstream timeslots including N<M upstream voice timeslots and a common upstream signaling timeslot.

5 40. The improved resource module of Claim 39 wherein the demultiplexer circuit includes an upstream signaling mapper for mapping signaling information from the common upstream signaling timeslot to the second signaling timeslots.

10 41. A communication system comprising:

a communication bus for carrying HDLC encapsulated Ethernet frames;
a subscriber resource module coupled to the communication bus for transmitting downstream HDLC encapsulated Ethernet frames to an external integrated access device and for receiving upstream HDLC encapsulated Ethernet frames from the external integrated access device;

15 an Ethernet protocol resource module including:

an Ethernet port for receiving downstream Ethernet frames from an external data network and for transmitting upstream Ethernet frames to the external data network;

an HDLC port for transmitting downstream HDLC encapsulated Ethernet frames to the communication bus and for receiving upstream HDLC encapsulated Ethernet frames from the communication bus; and

20 a protocol converter for converting downstream Ethernet frames to downstream HDLC encapsulated Ethernet frames and for converting upstream HDLC encapsulated Ethernet frames to upstream Ethernet frames.

25

1/58

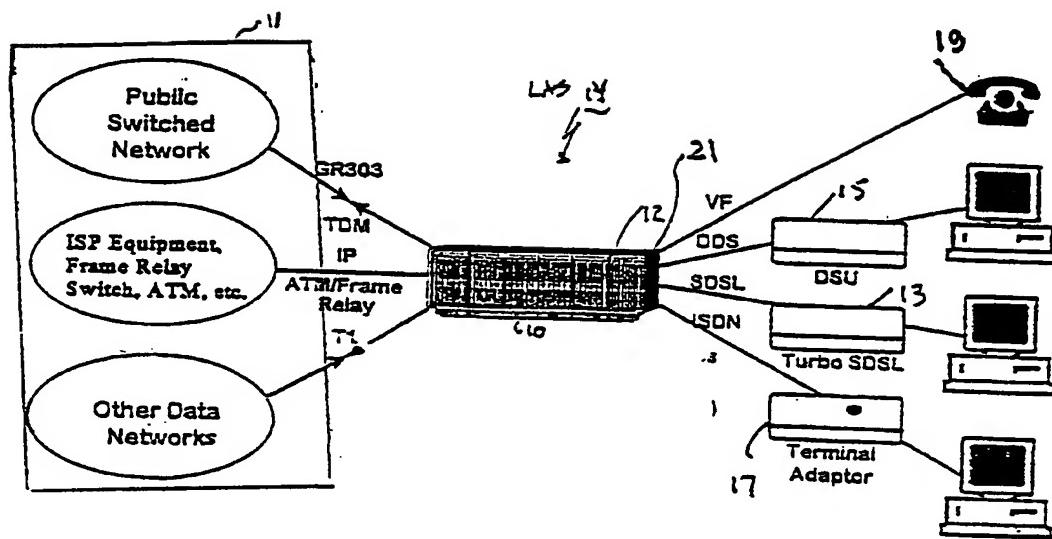


Fig. 1

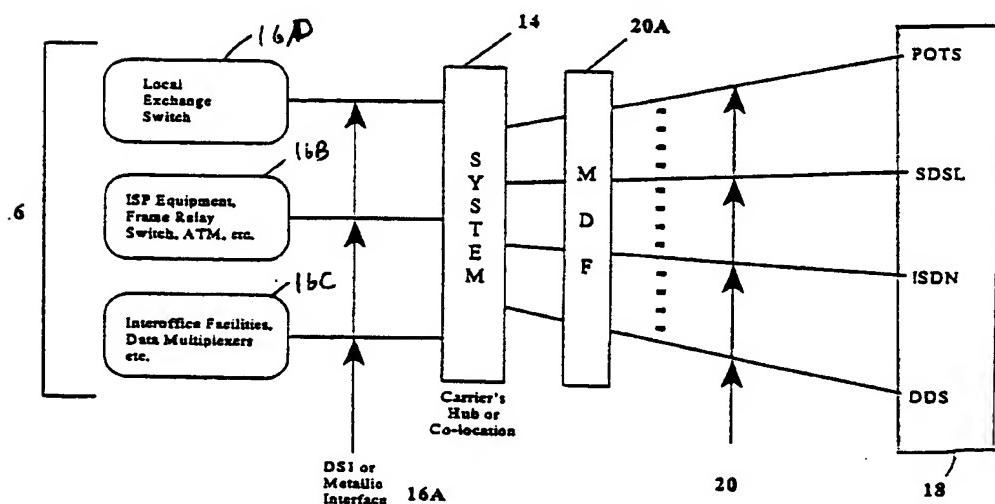


Fig. 2

2/58

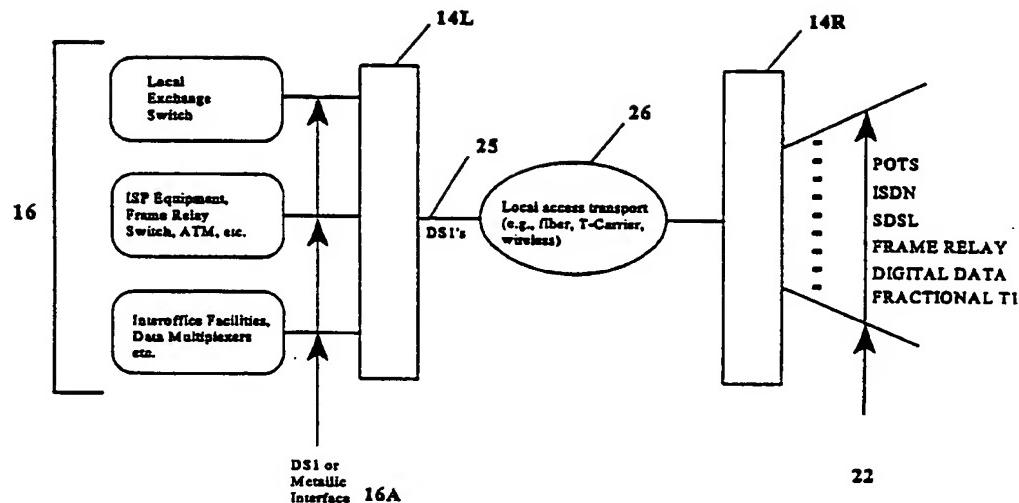


Fig. 3

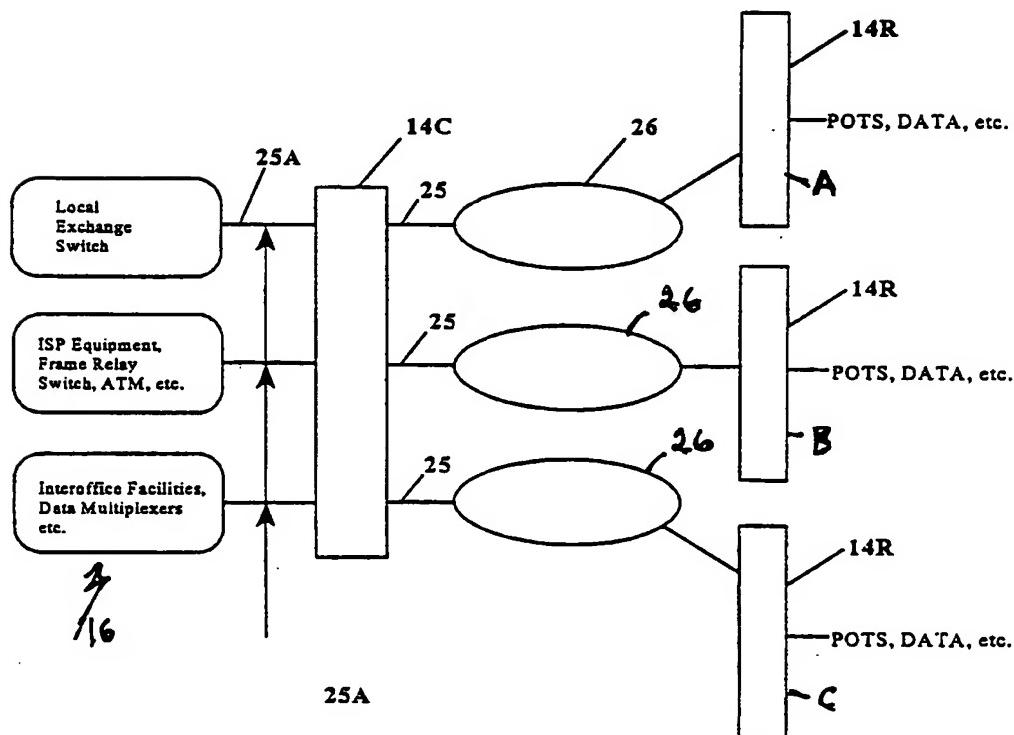


Fig. 4

3 / 58

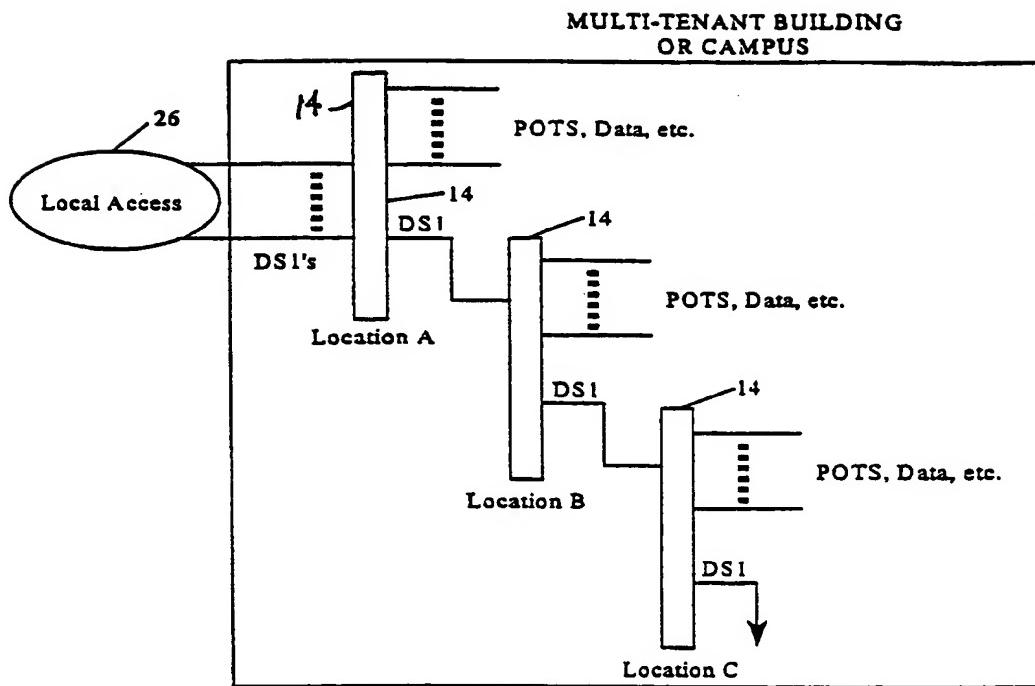


Fig. 5

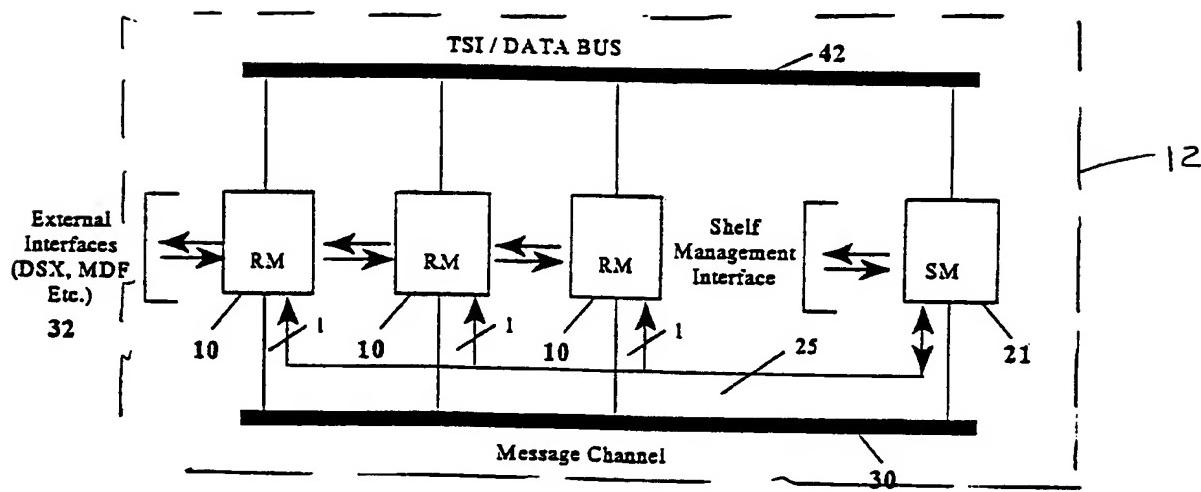


Fig. 6

4 | 58

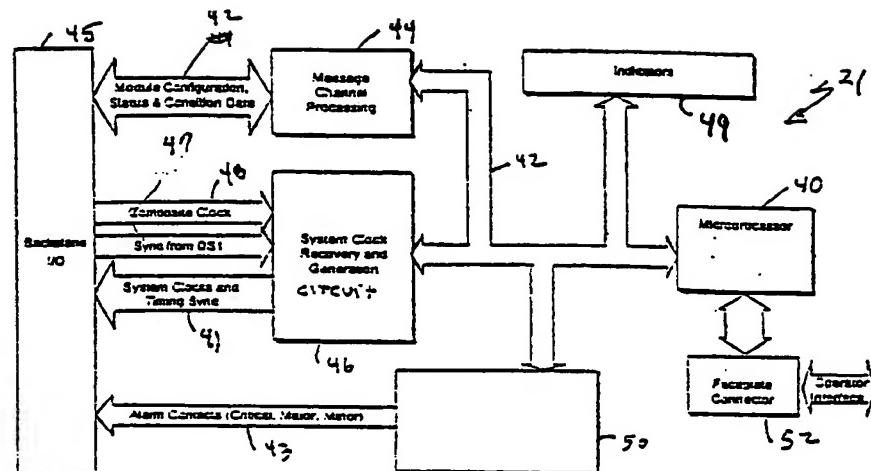


Fig. 7

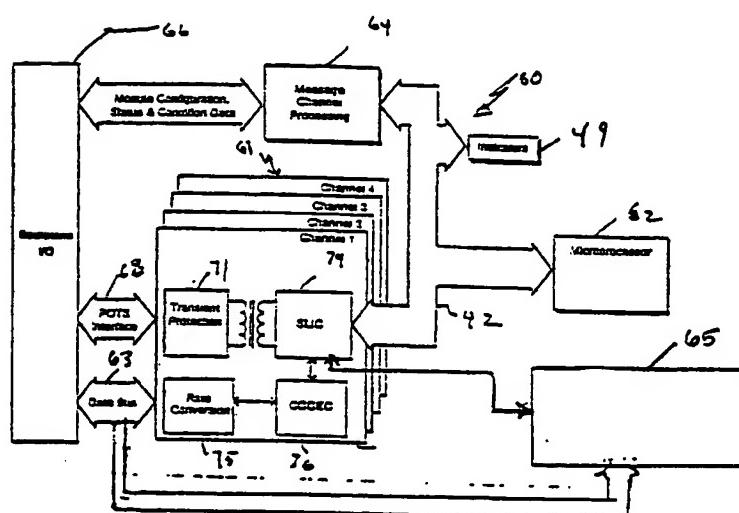


Fig. 8

5/58

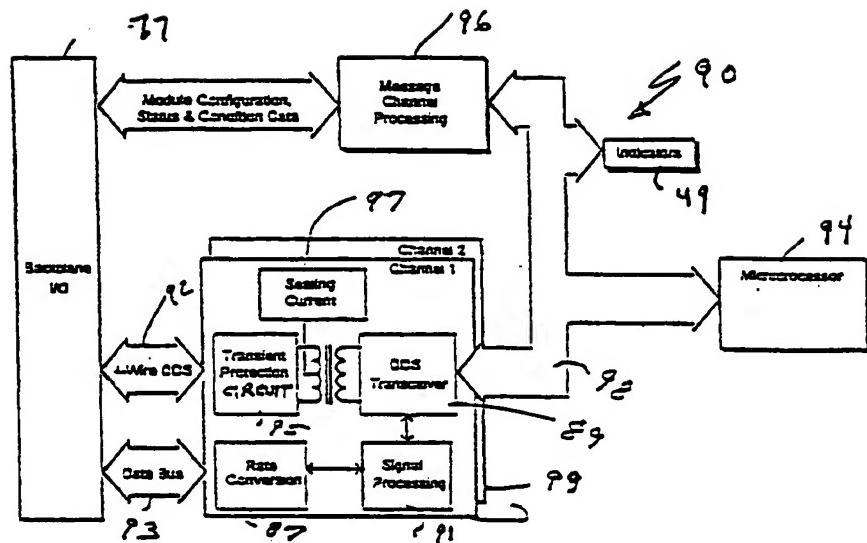


Fig. 9

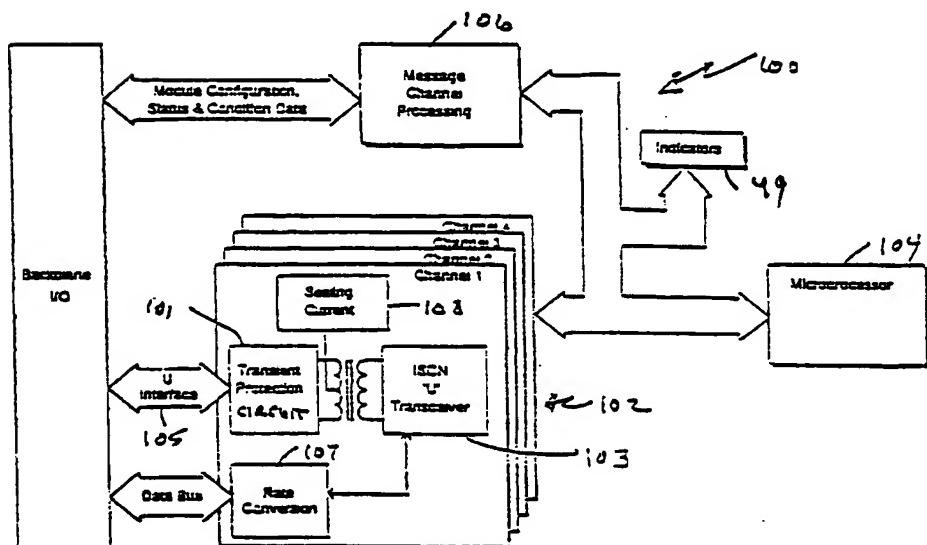


Fig. 10

6 / 58

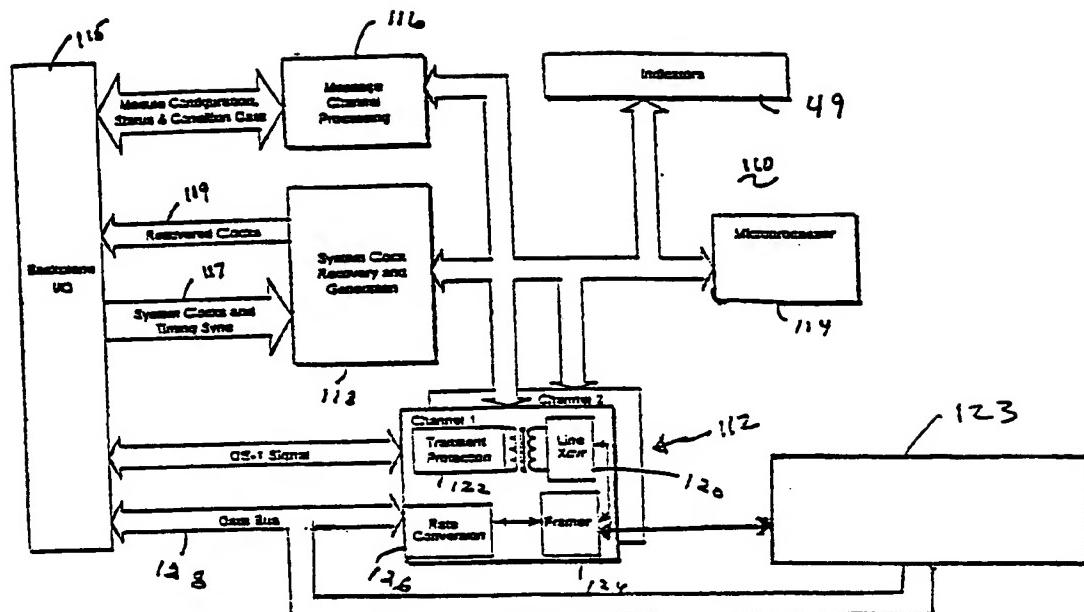


Fig. 11

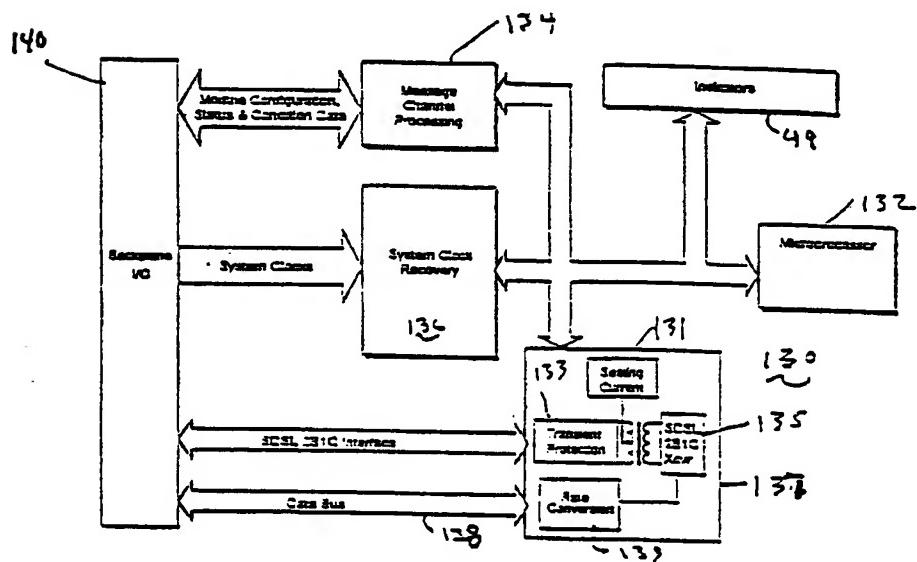


Fig. 12

7/58

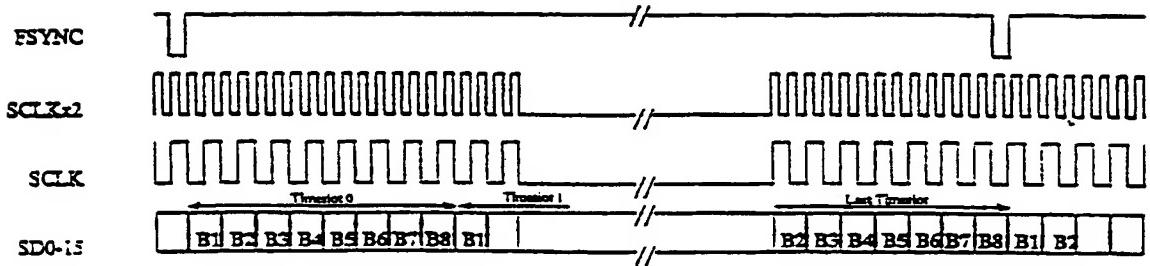


Fig. 13

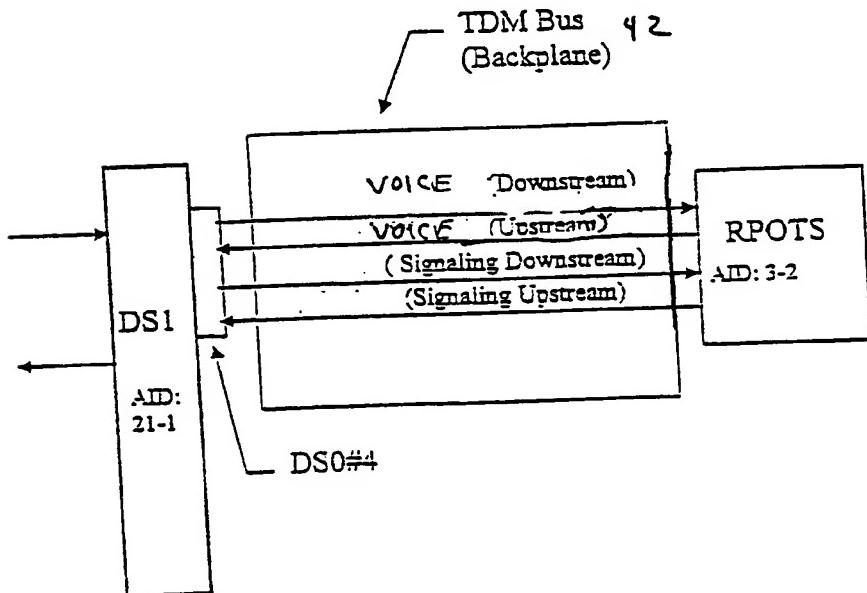


Fig. 14

8 / 58

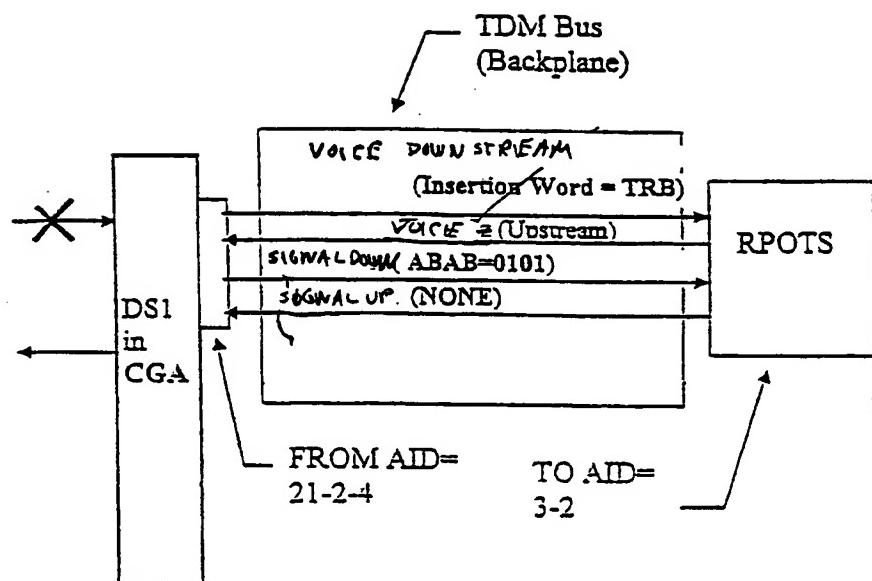


Fig. 15

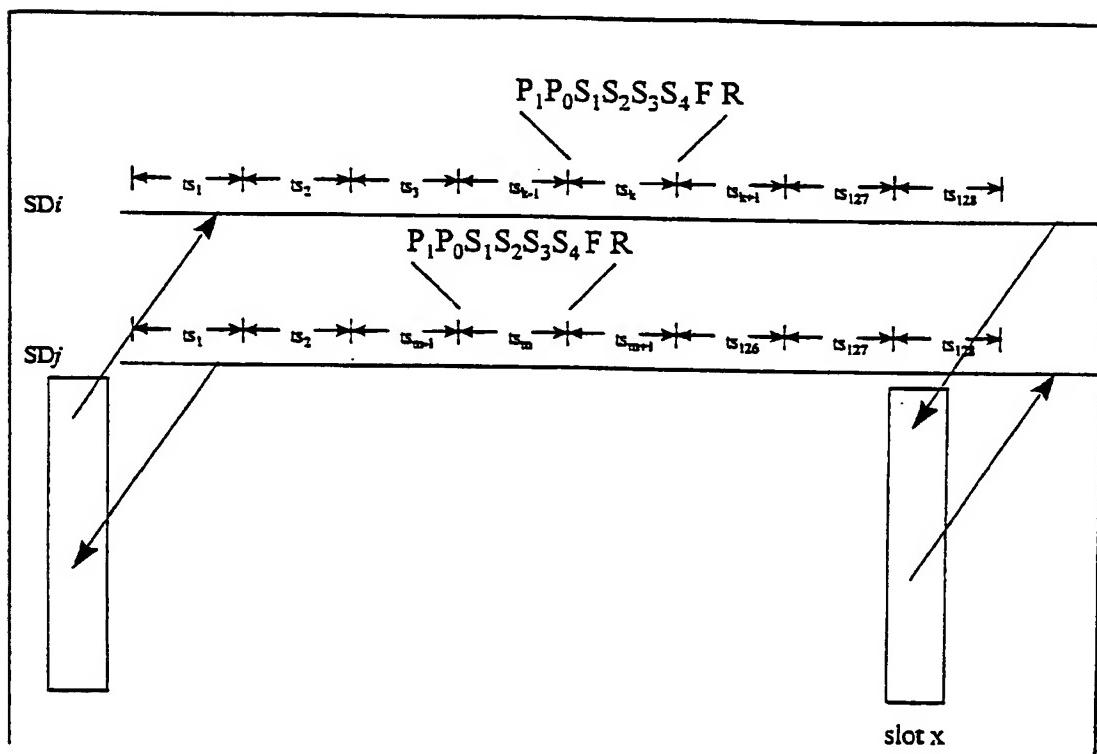


Fig. 16

3/58

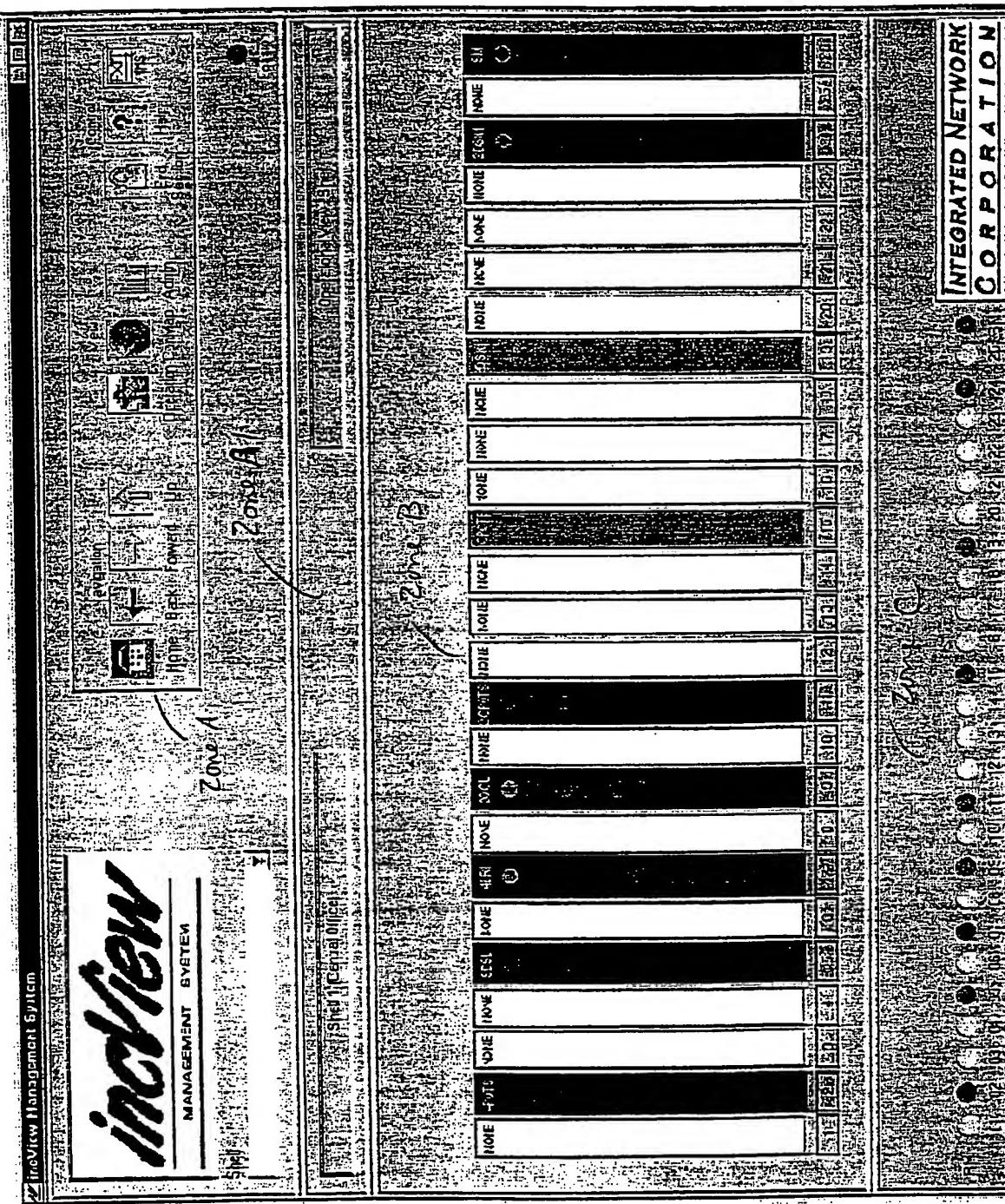


Fig. 17

10/58

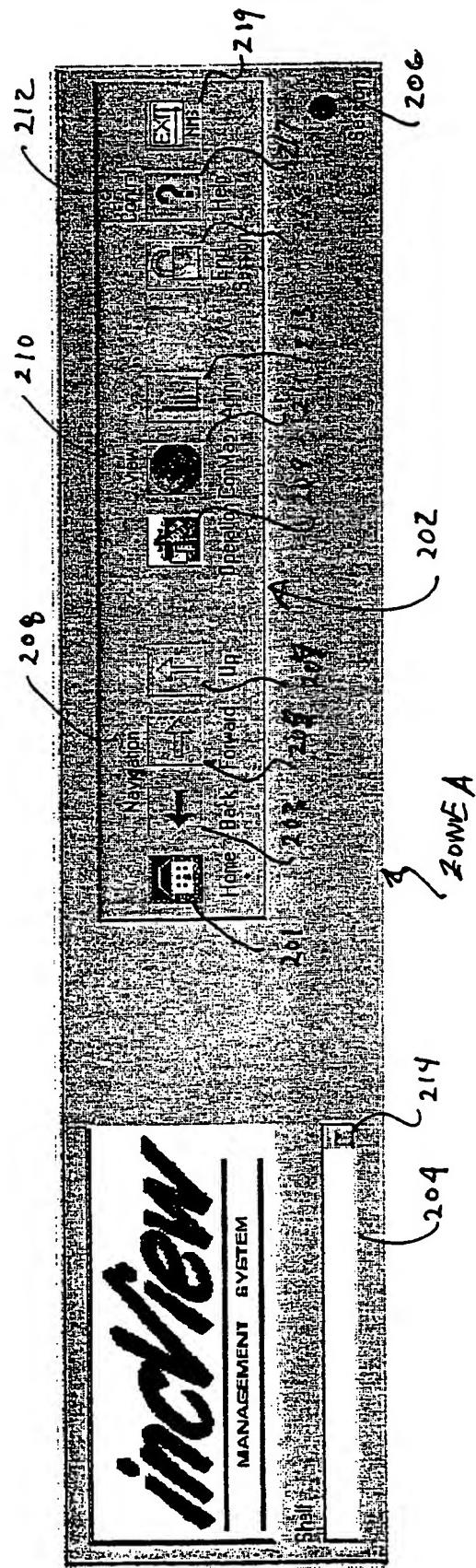


Fig. 18

11/58



F16-19

12 / 58

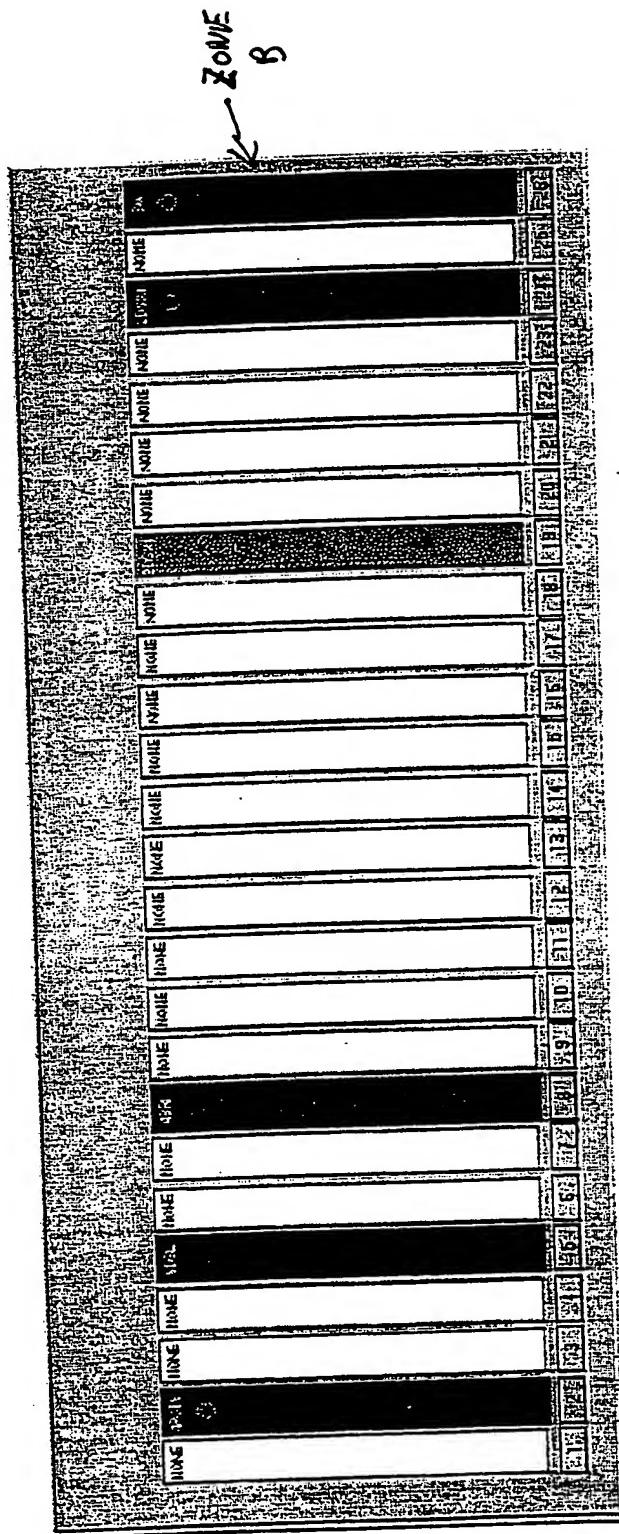
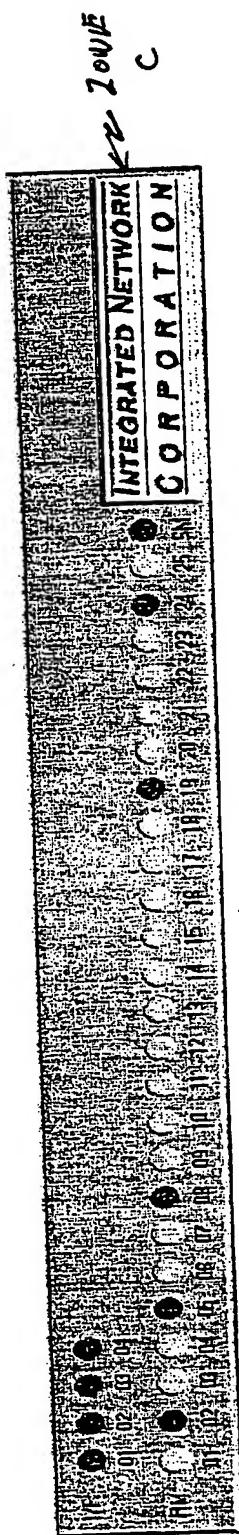


FIG. 20

13/58



FILED 21

14 | 58

214

30

232

F 16. 22

15/58

130

SM [Shell Manager]	125F40101
NO ID	Primary
OFFICE	INTERNAL
APPLY	CANCEL

Fig. 23

16|58

Order of Ven	
Eq Type	SDS
Eq Ver	226.D_05
Channels	VONE (0)
Installed Eq Type	SDS
APPLY	CANCEL

01



17 | 58

Start	Ending	Ever	Port	Main
Data Rate: 768 kbps				
Setting Current: DN				
<input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="OK"/> <input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="APPLY"/> <input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="CANCEL"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="OK"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Cancel"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Close"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Print"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Help"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Exit"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Print"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Help"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Exit"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Print"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Help"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Exit"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Print"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Help"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Exit"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Print"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Help"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Exit"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Print"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Help"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Exit"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Print"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Help"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Exit"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Print"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Help"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Exit"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Print"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Help"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Exit"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Print"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Help"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Exit"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Print"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Help"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Exit"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Print"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Help"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Exit"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Print"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Help"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Exit"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Print"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Help"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Exit"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Print"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Help"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Exit"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Print"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Help"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Exit"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Print"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Help"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Exit"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Print"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Help"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Exit"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Print"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Help"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Exit"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Print"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Help"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Exit"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Print"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Help"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Exit"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Print"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Help"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Exit"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Print"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Help"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Exit"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Print"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Help"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Exit"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Print"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Help"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Exit"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Print"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Help"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Exit"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Print"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Help"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Exit"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Print"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Help"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Exit"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Print"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Help"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Exit"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Print"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Help"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Exit"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Print"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Help"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Exit"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Print"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Help"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Exit"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Print"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Help"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Exit"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Print"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Help"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Exit"/>				
<input style="width: 100px; height: 30px; margin-left: 10px; margin-bottom: 5px; font-size: 10pt; border-radius: 10px; border: 1px solid black; background-color: #f0f0f0; color: black; font-weight: bold; padding: 5px;" type="button" value="Print"/>				
<input type="button" value="Help" style="width: 100px; height: 30px;				

230 1242

18 | 58

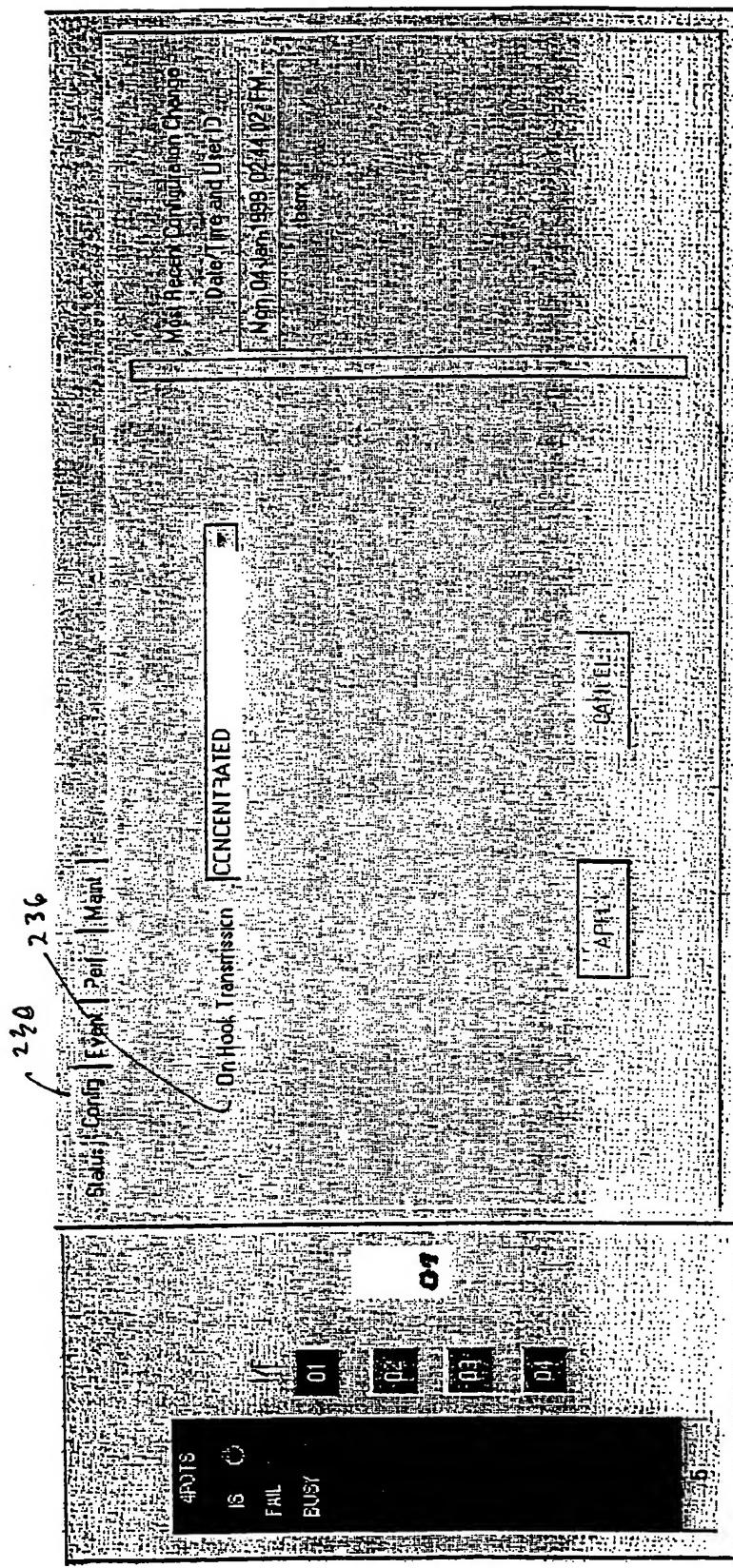


Fig. 26

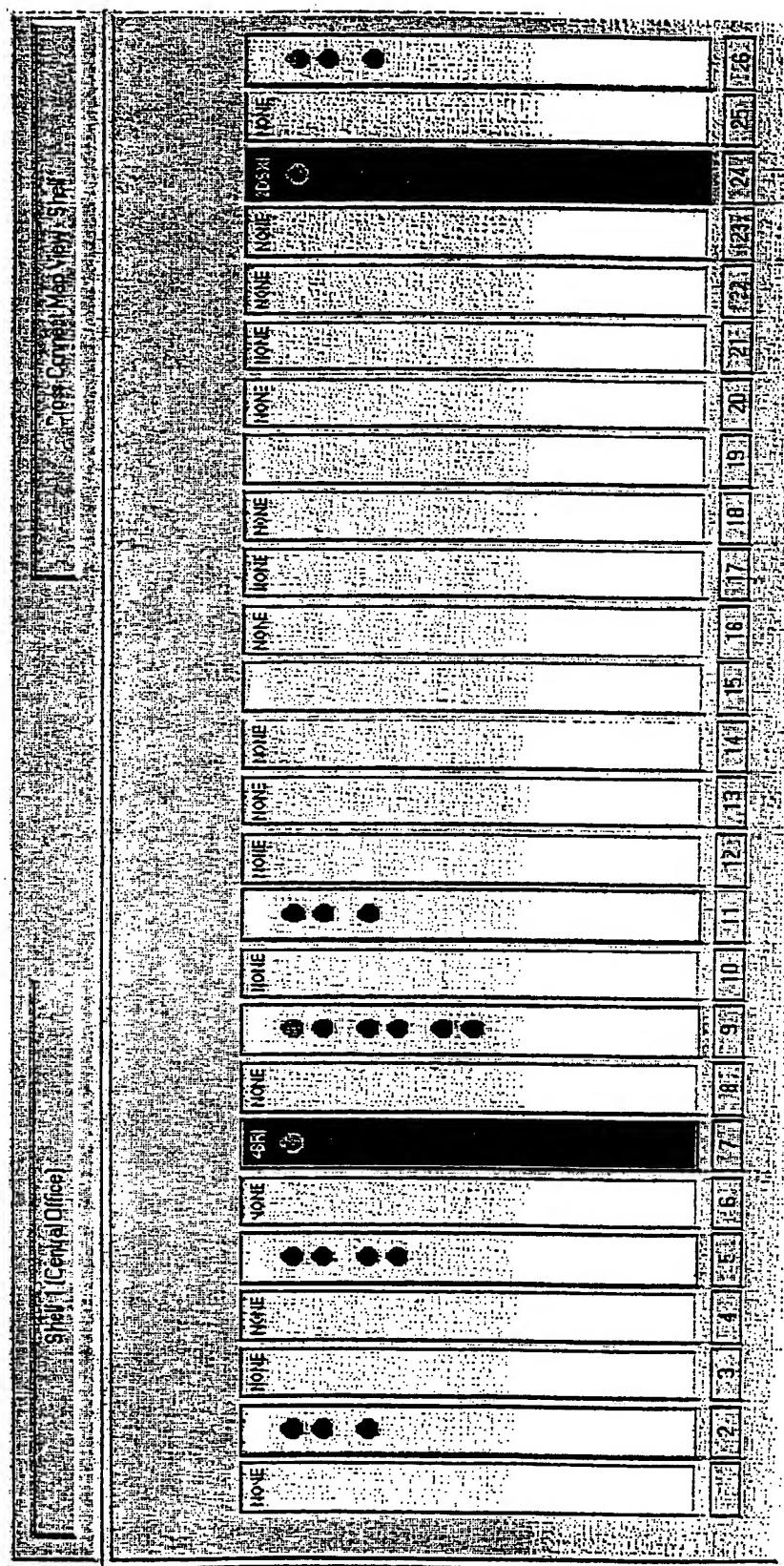
19158

Feb. 27

20158

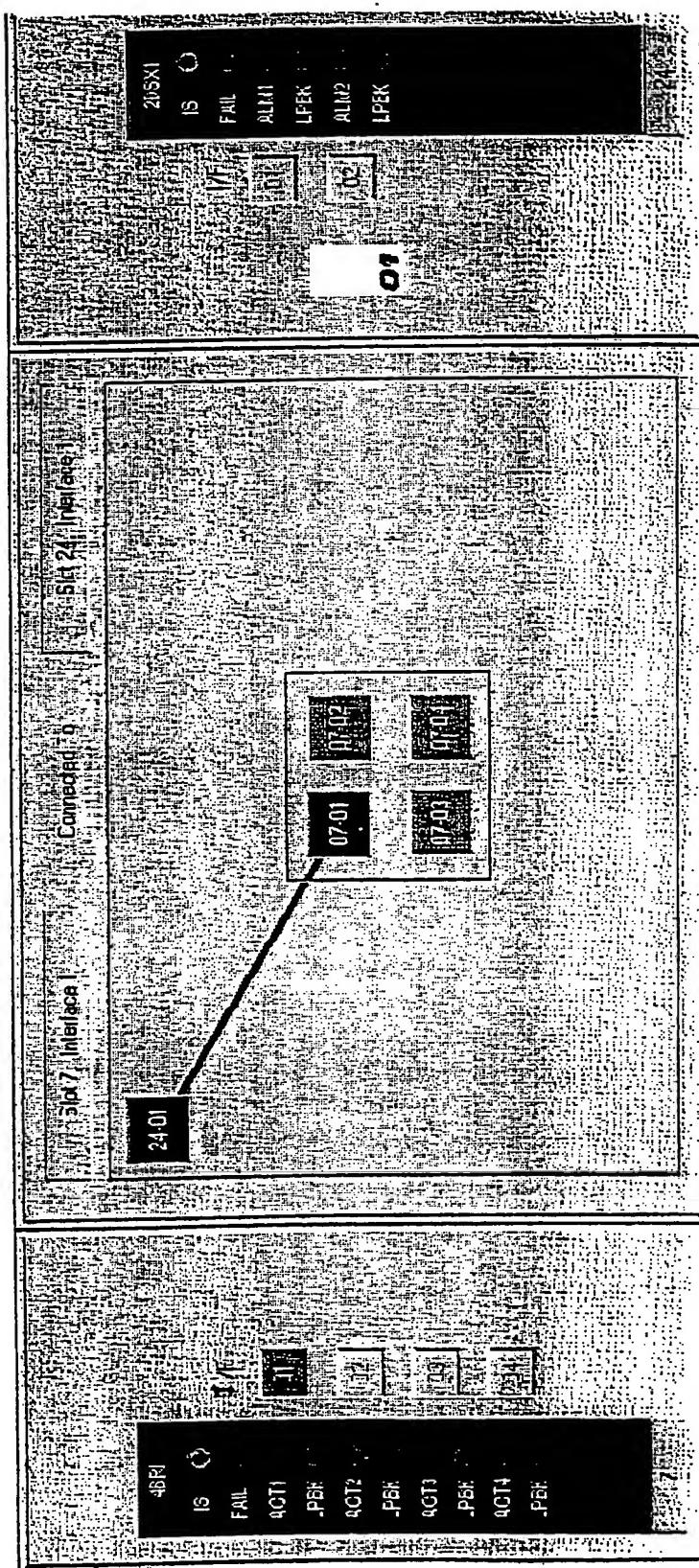
Fig. 28

21 | 58



T-16. 29

22/58



23158

Sheet 1 Central Diesel Site 7 Initiative		Zone E		Zone F	
Tires of TIS Map		Front TS		Rear TS	
16	FAIL	01	02	03	04
ACT1	LPER1	ACT2	LPER2	ACT3	LPER3
ACT4	LPER4				

P6 / 58

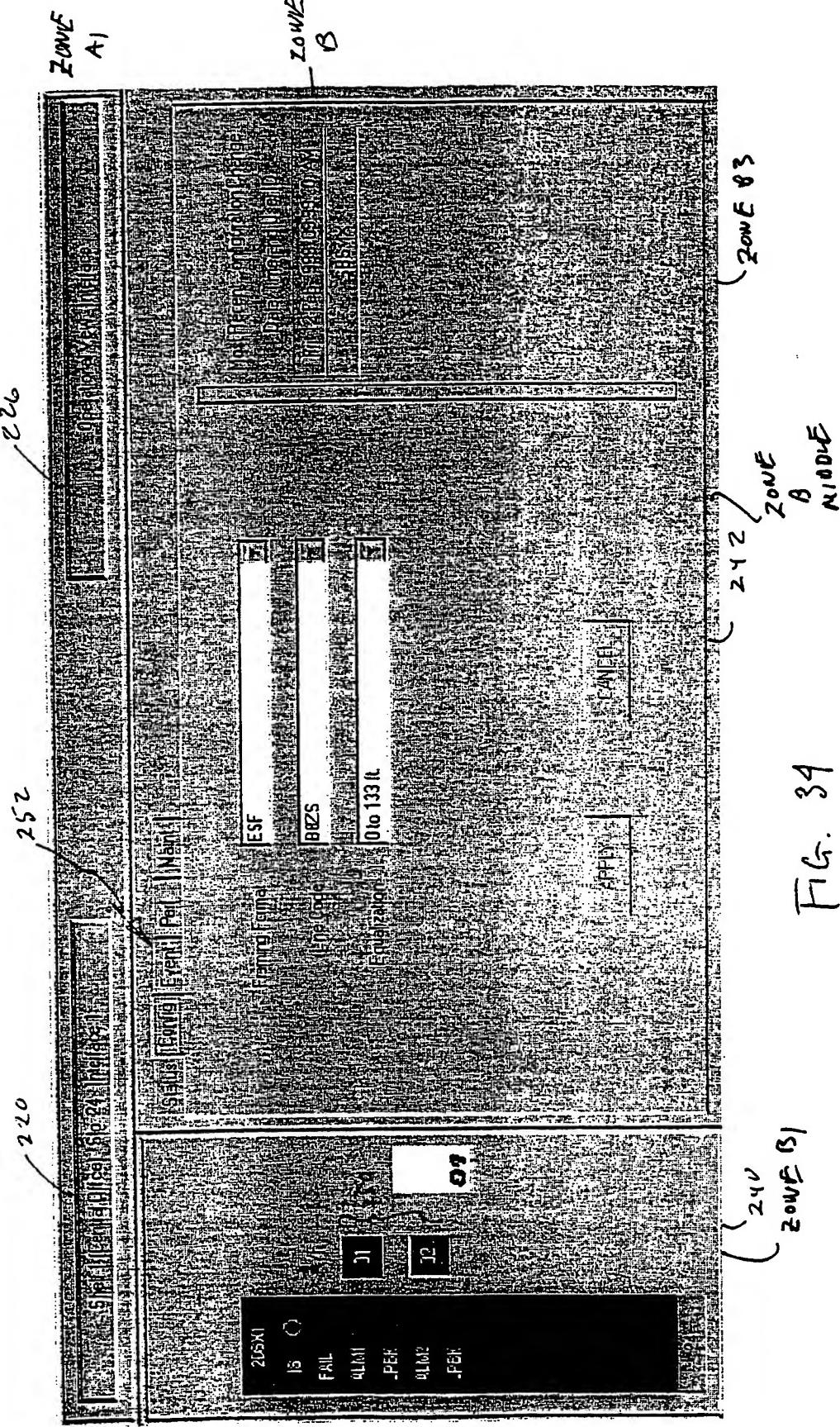


Fig. 34

(ZONE 81)

242 / ZONE
B N100E

(ZONE 83)

27158

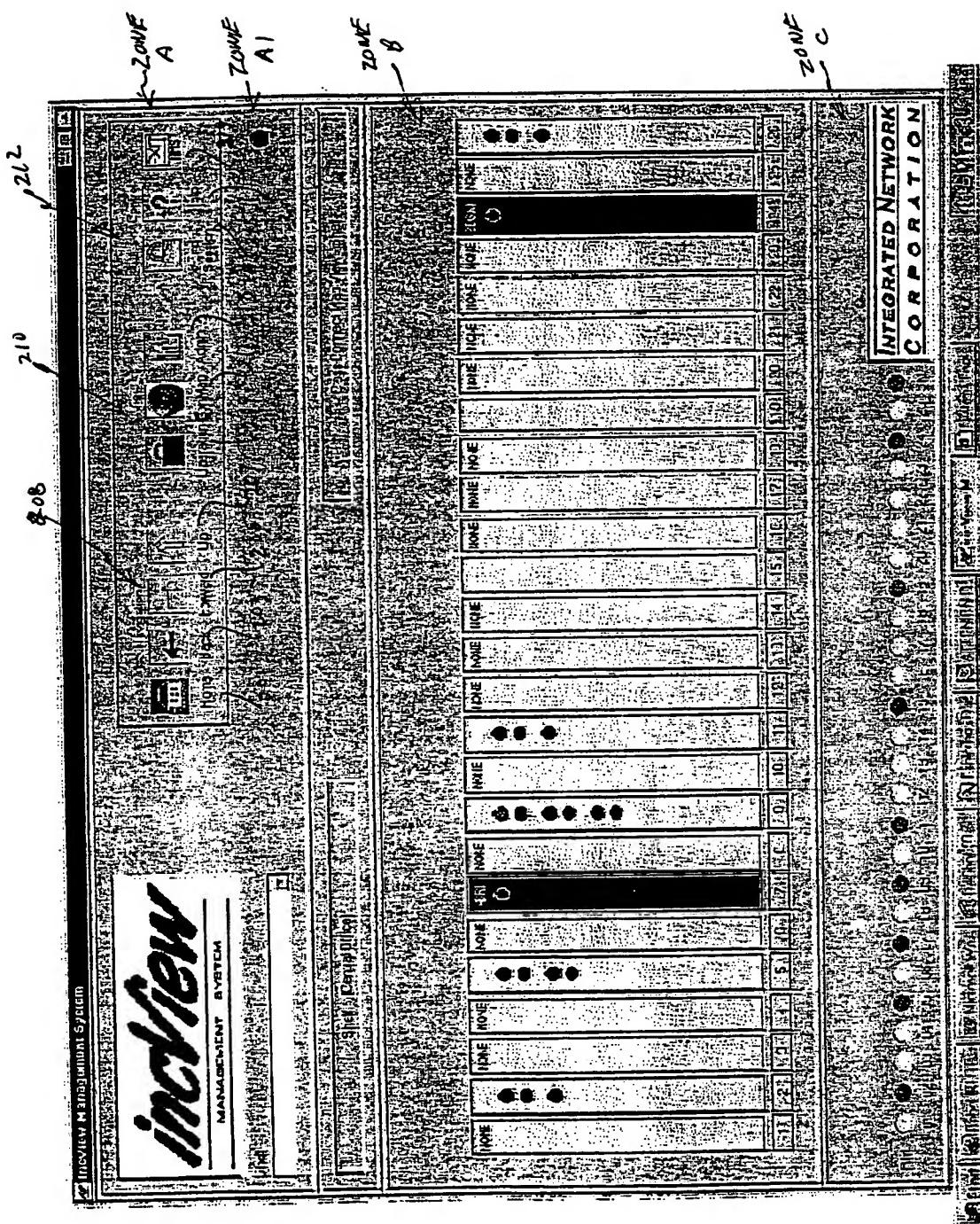
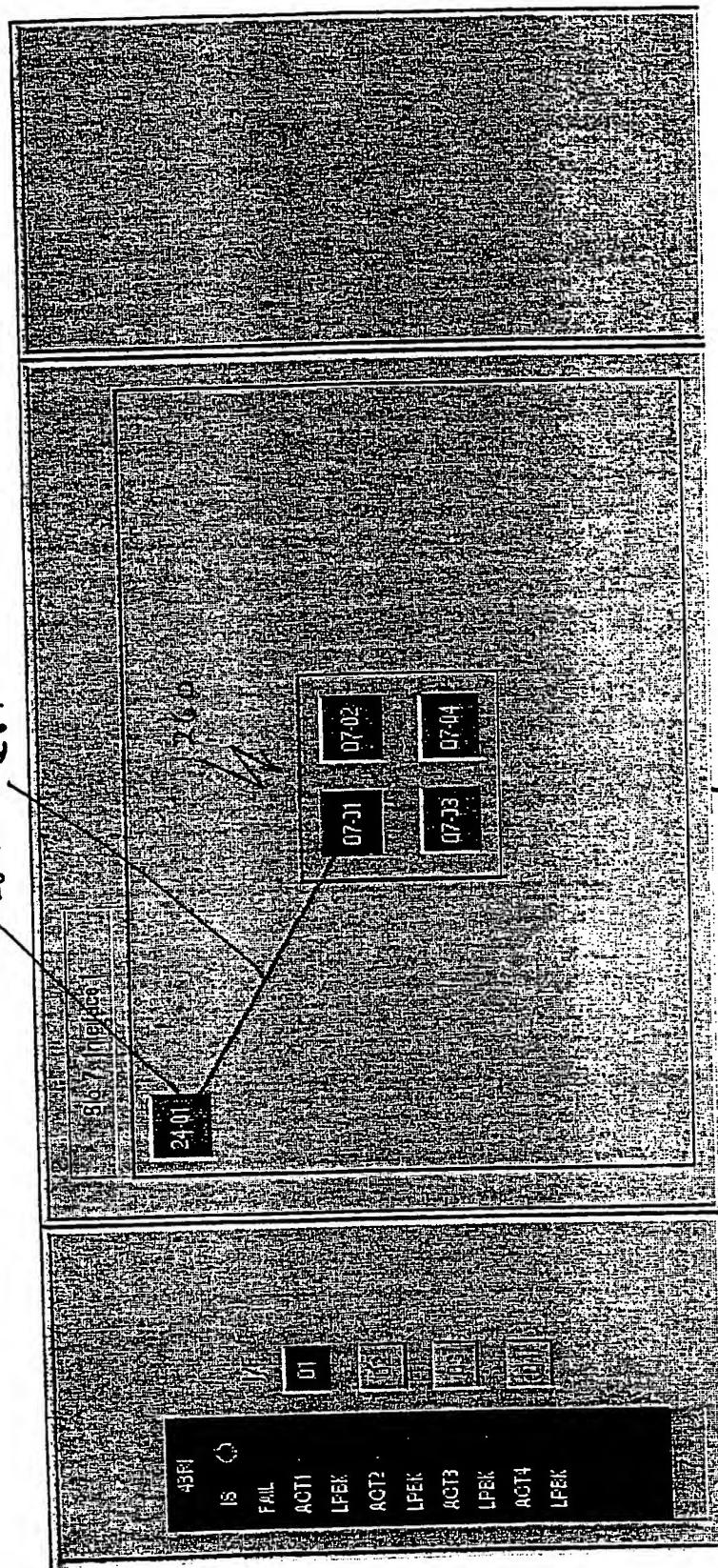


FIG. 35

28158



ZONE B (NODF)

ZONE B 1

Fig. 36

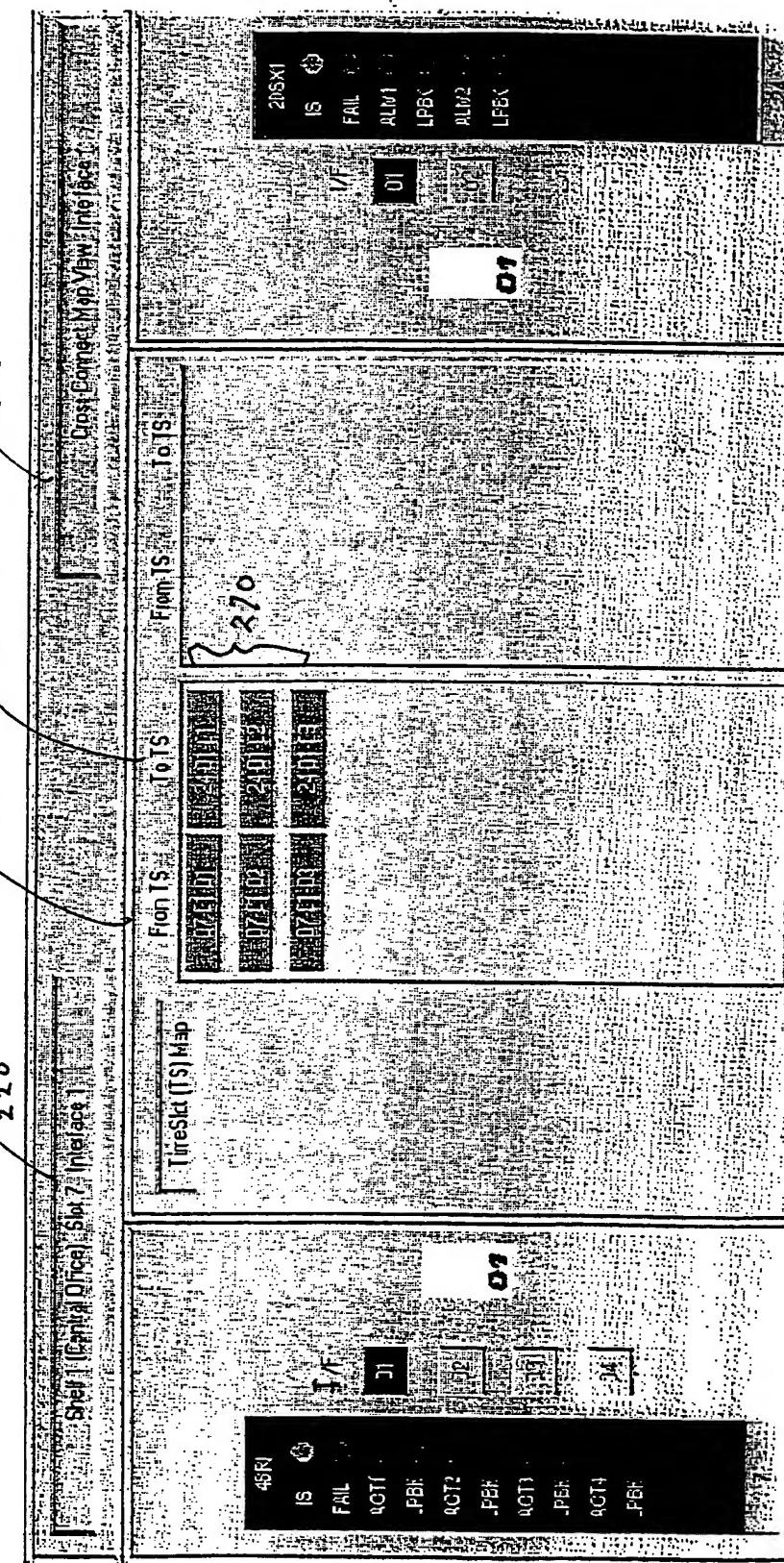
23/58

220

266

220

268



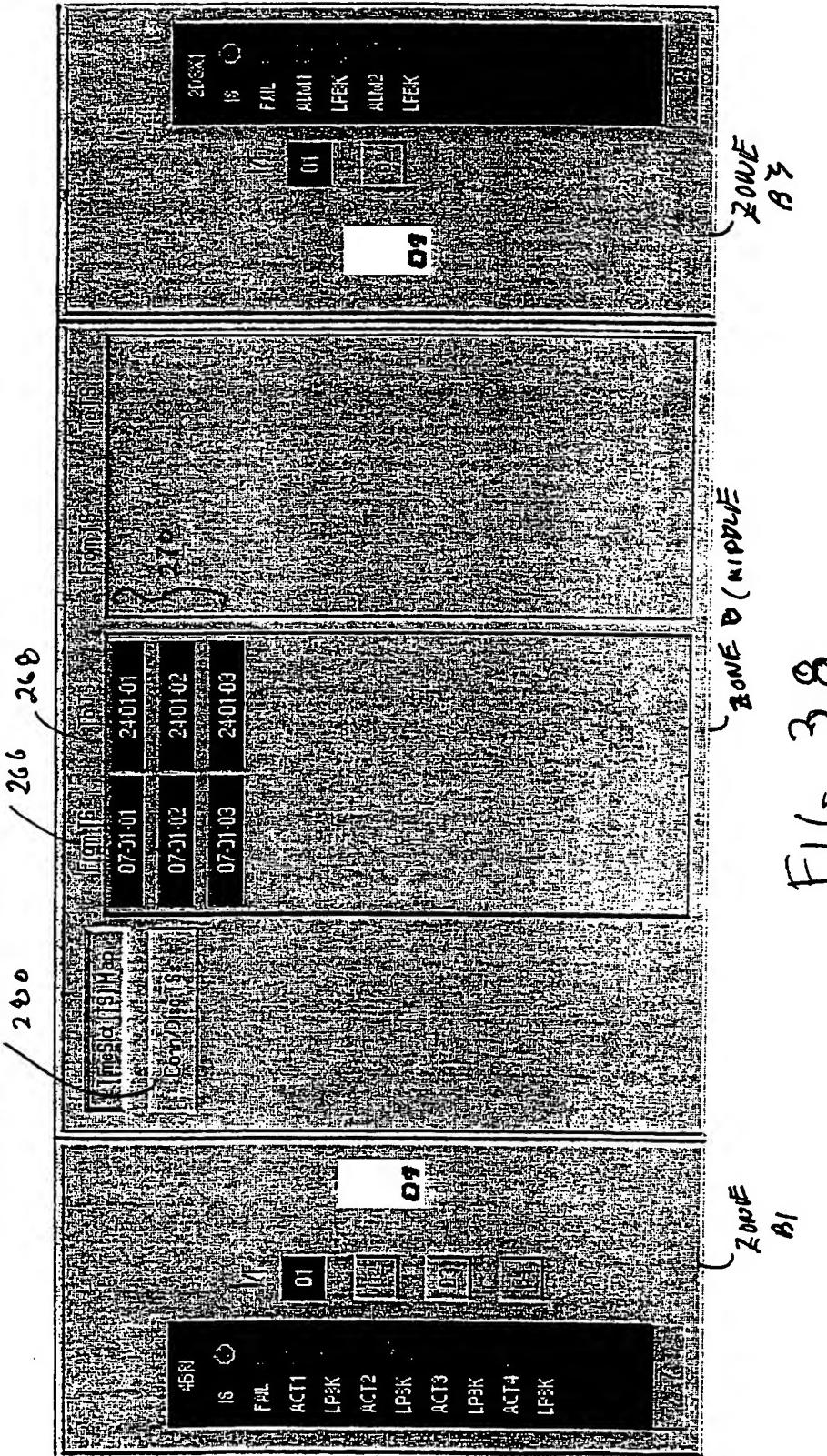
200E B3

200E B (NIDUE)

ZONE 81

T1C, 32.7

30158



31 | 58

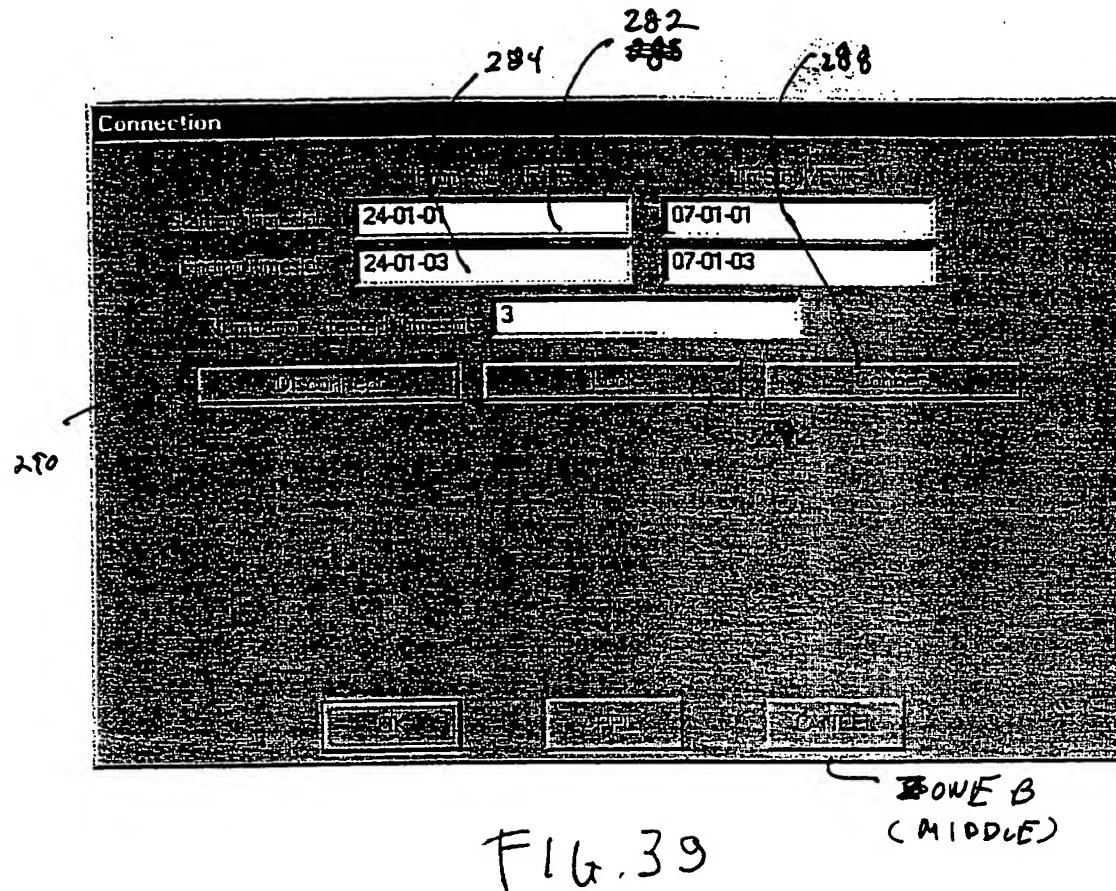
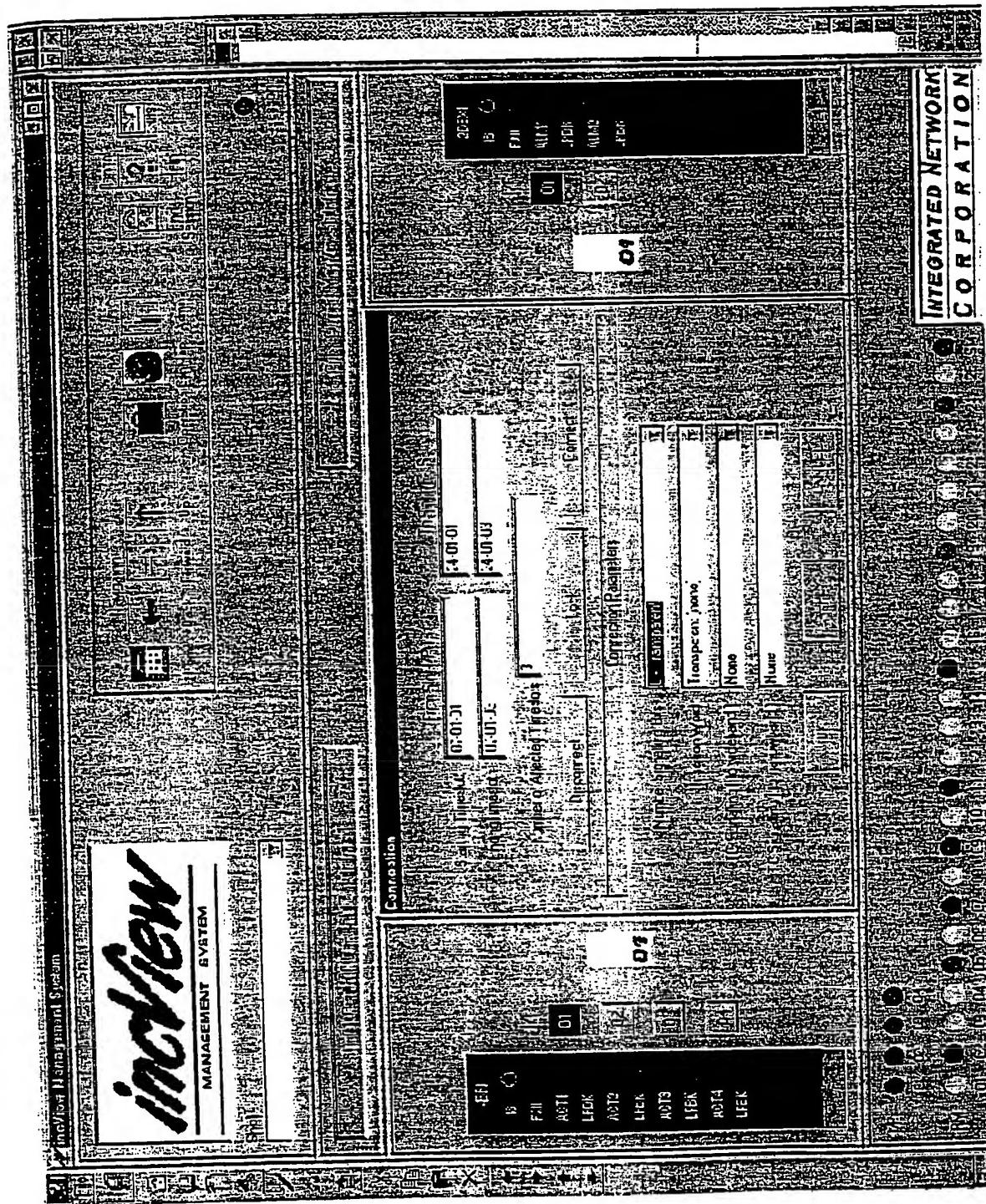


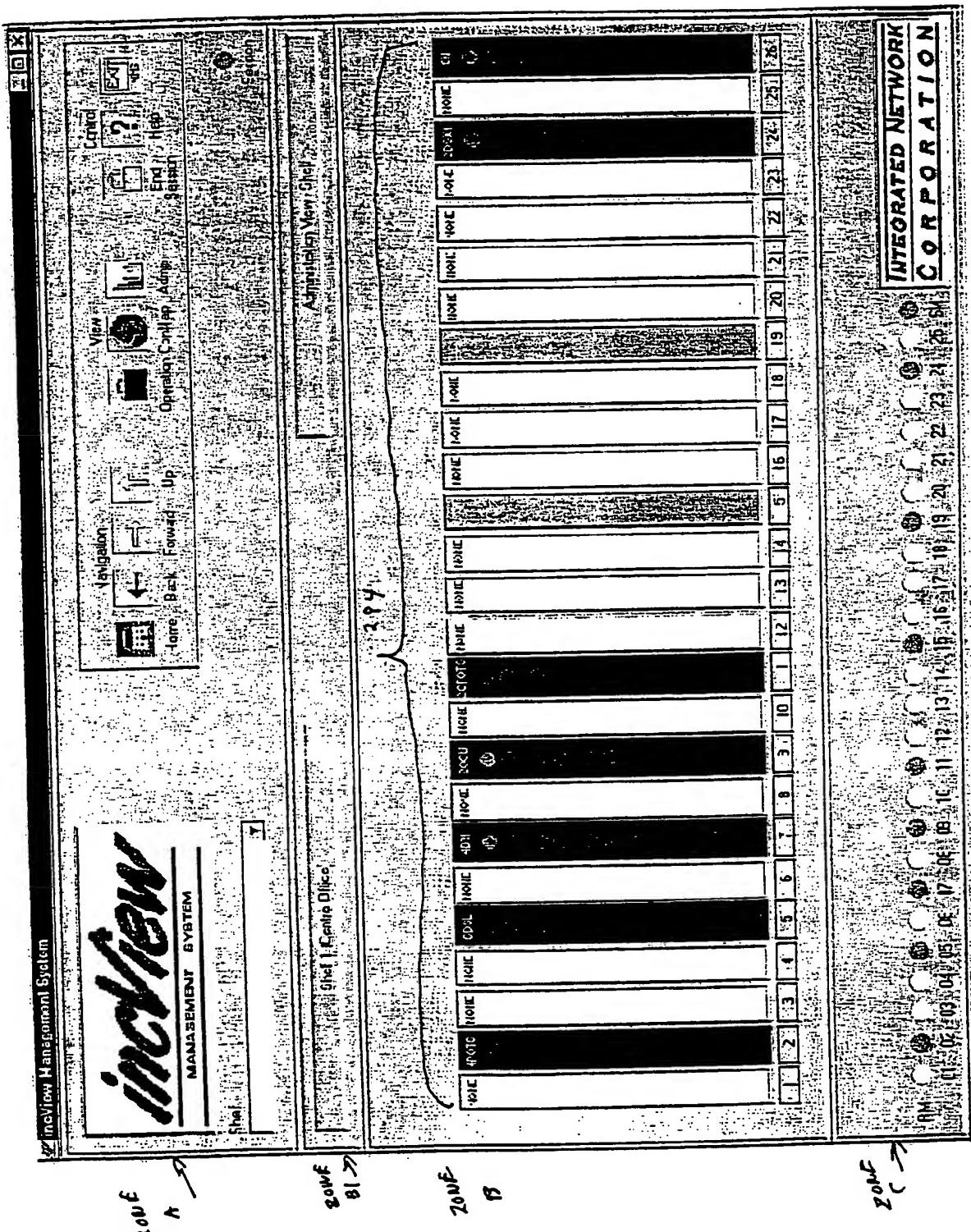
FIG. 39

32158



F16.4.φ

33 | 58



File 45

34158

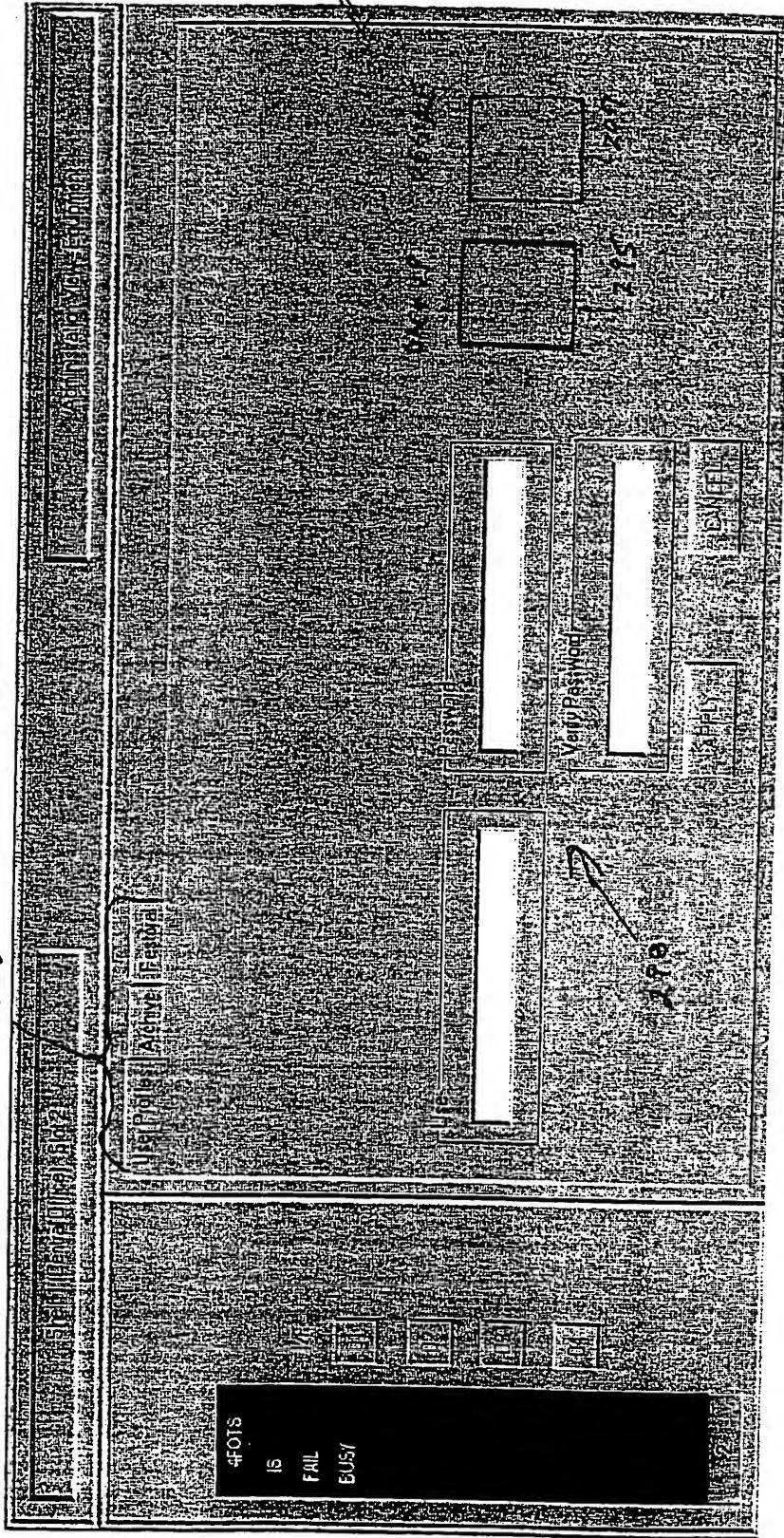
201/E
G

Fig A2

35 | 58

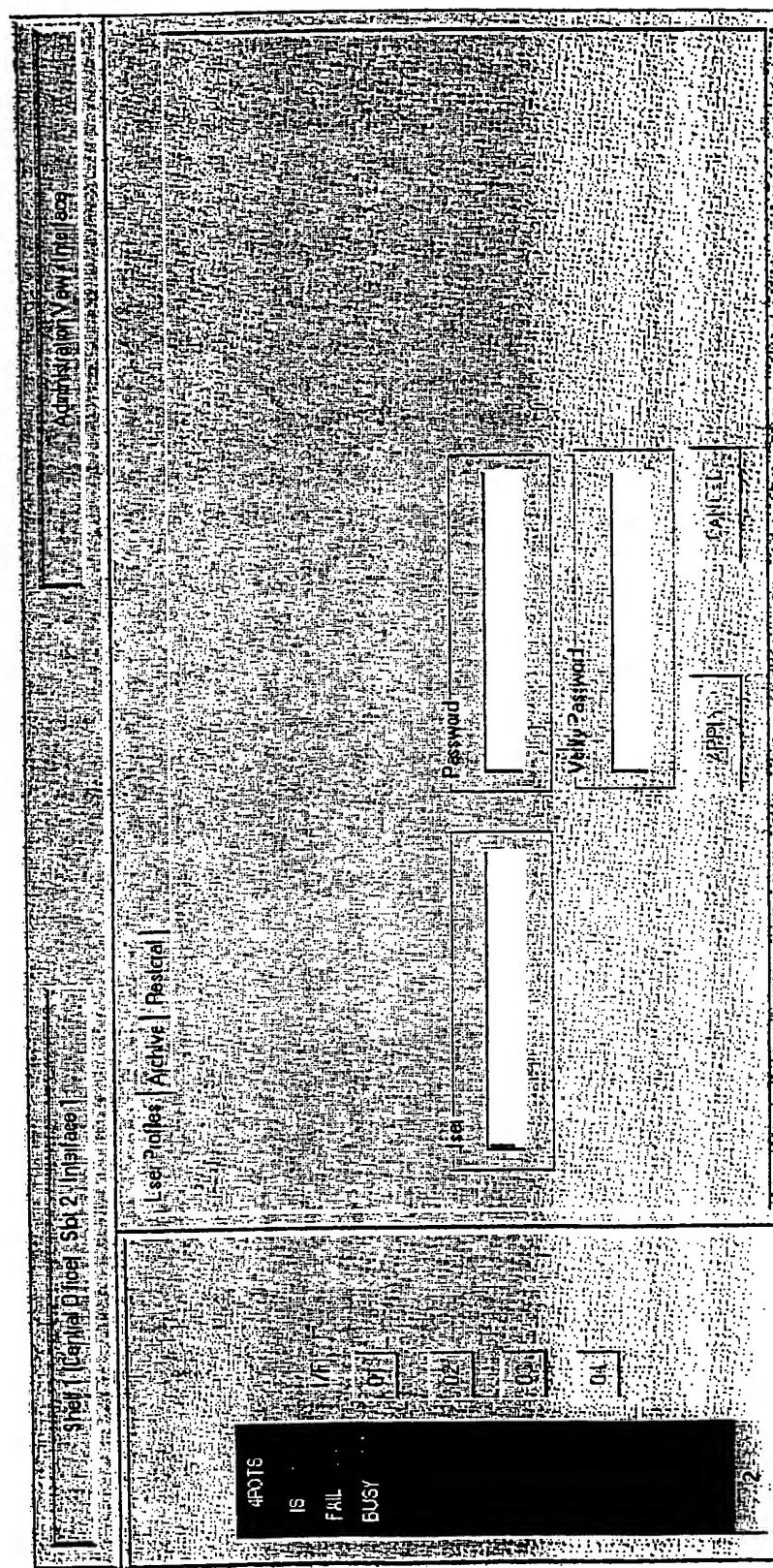


Fig. 43

36/58

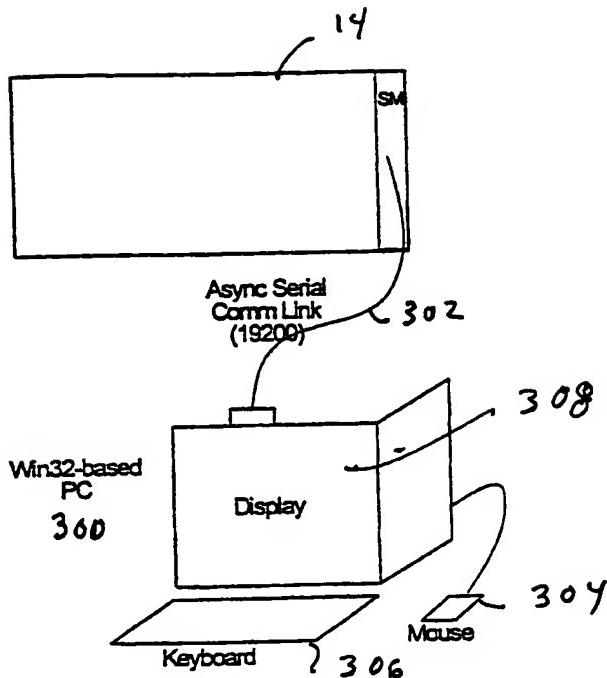


FIG. 44

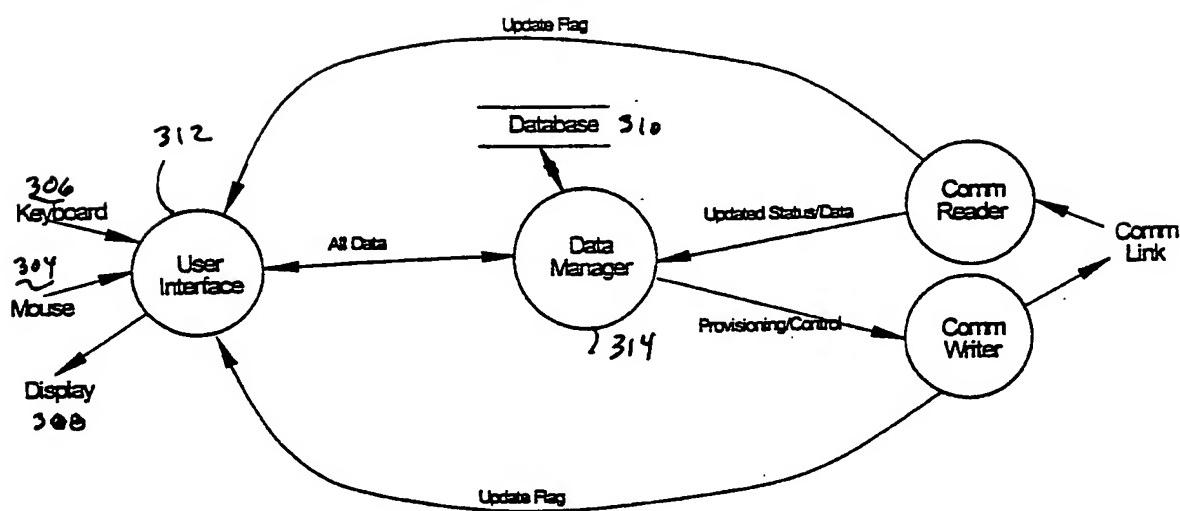


FIG. 45

37158

5

10

15

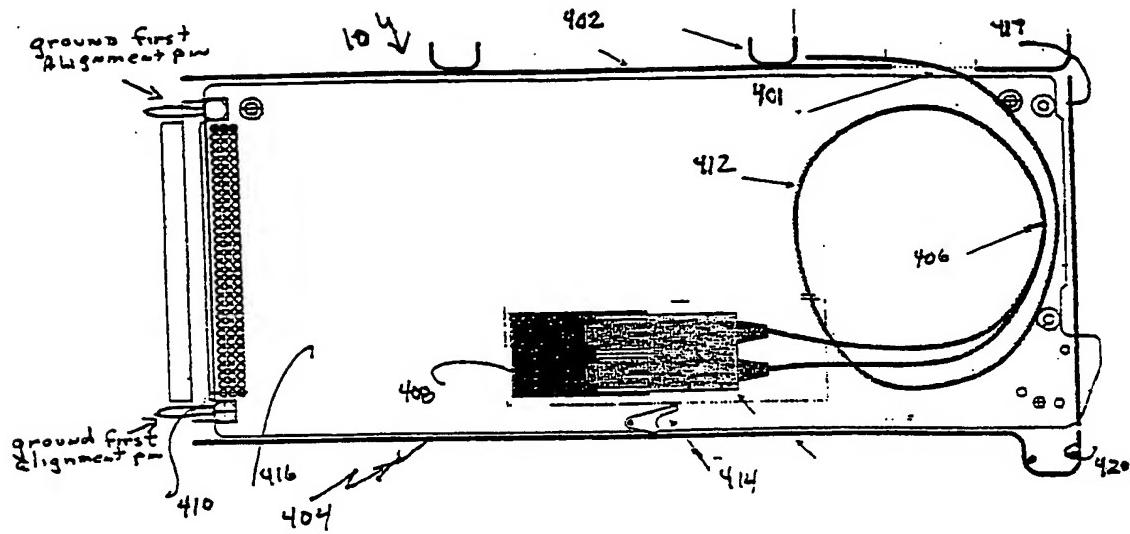


Fig. 46

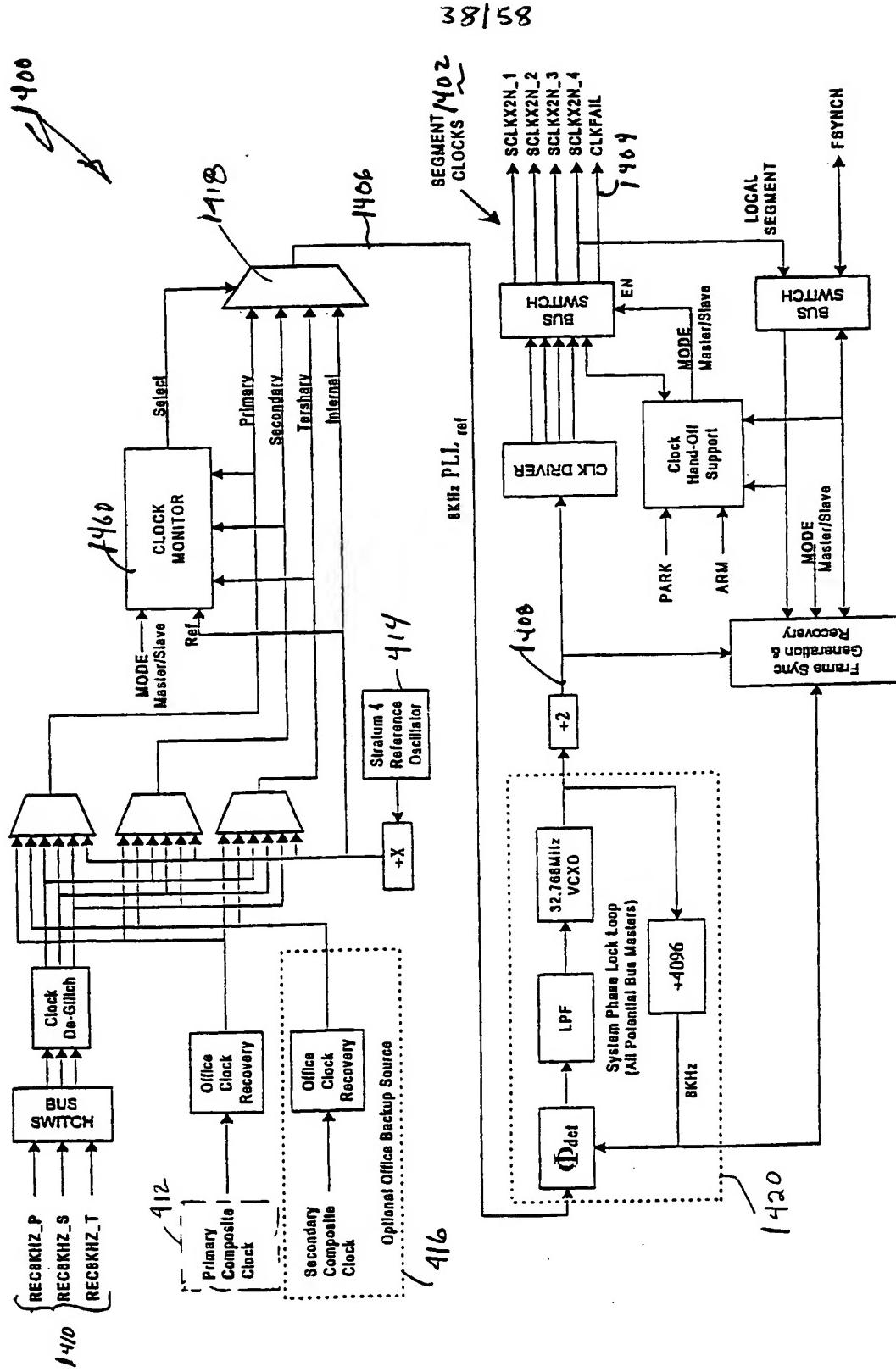


Fig. 47

39 | 58

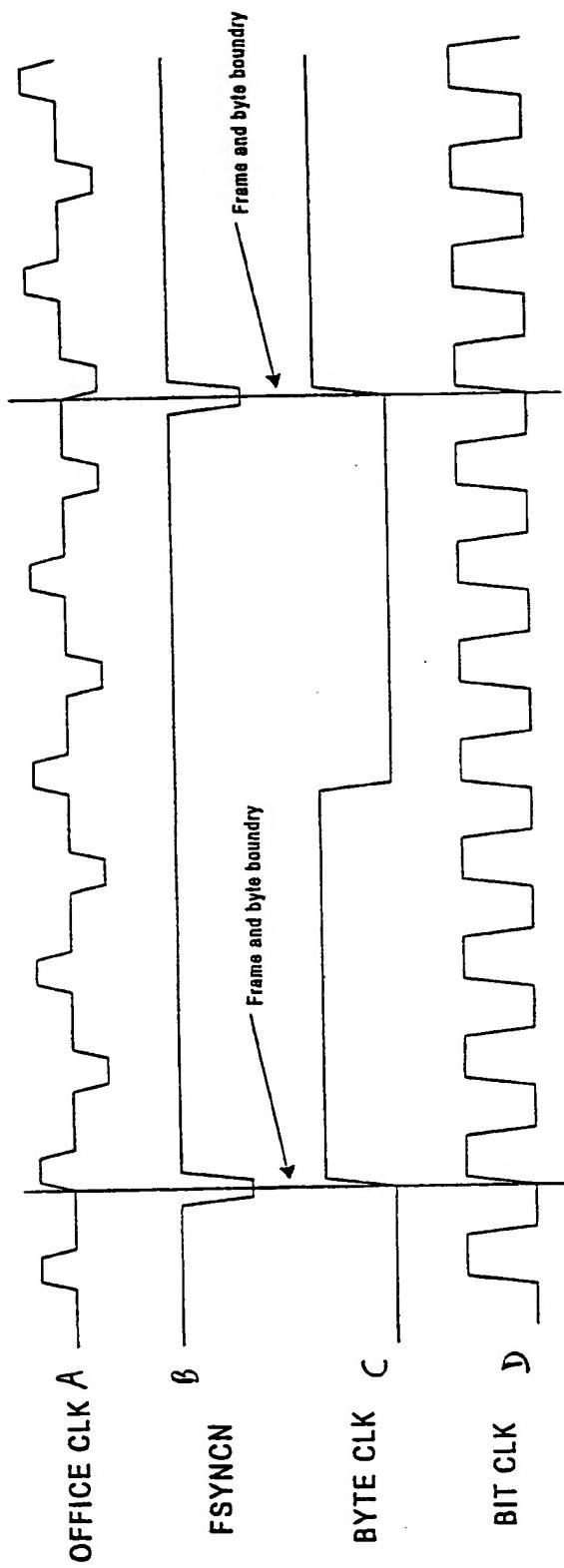


Fig. 4B

40/58

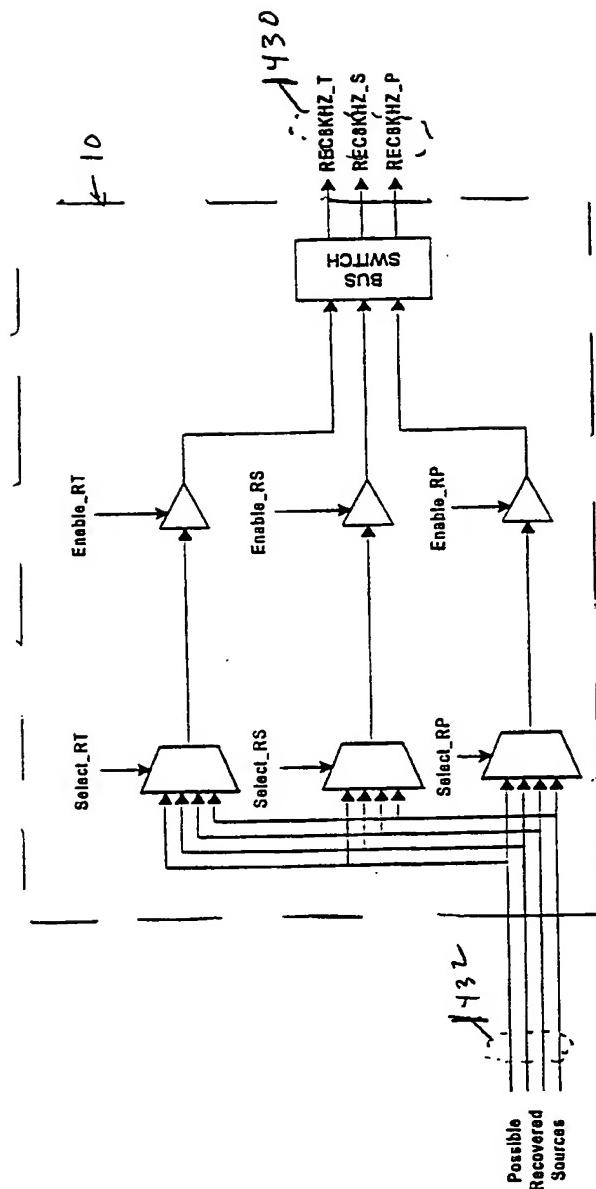


Fig. 49

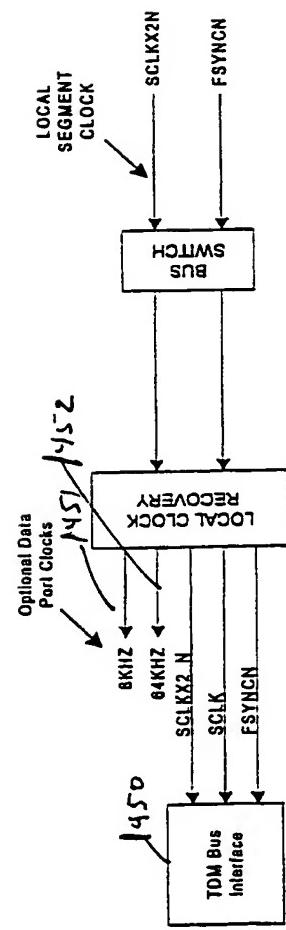


Fig. 50

41/58

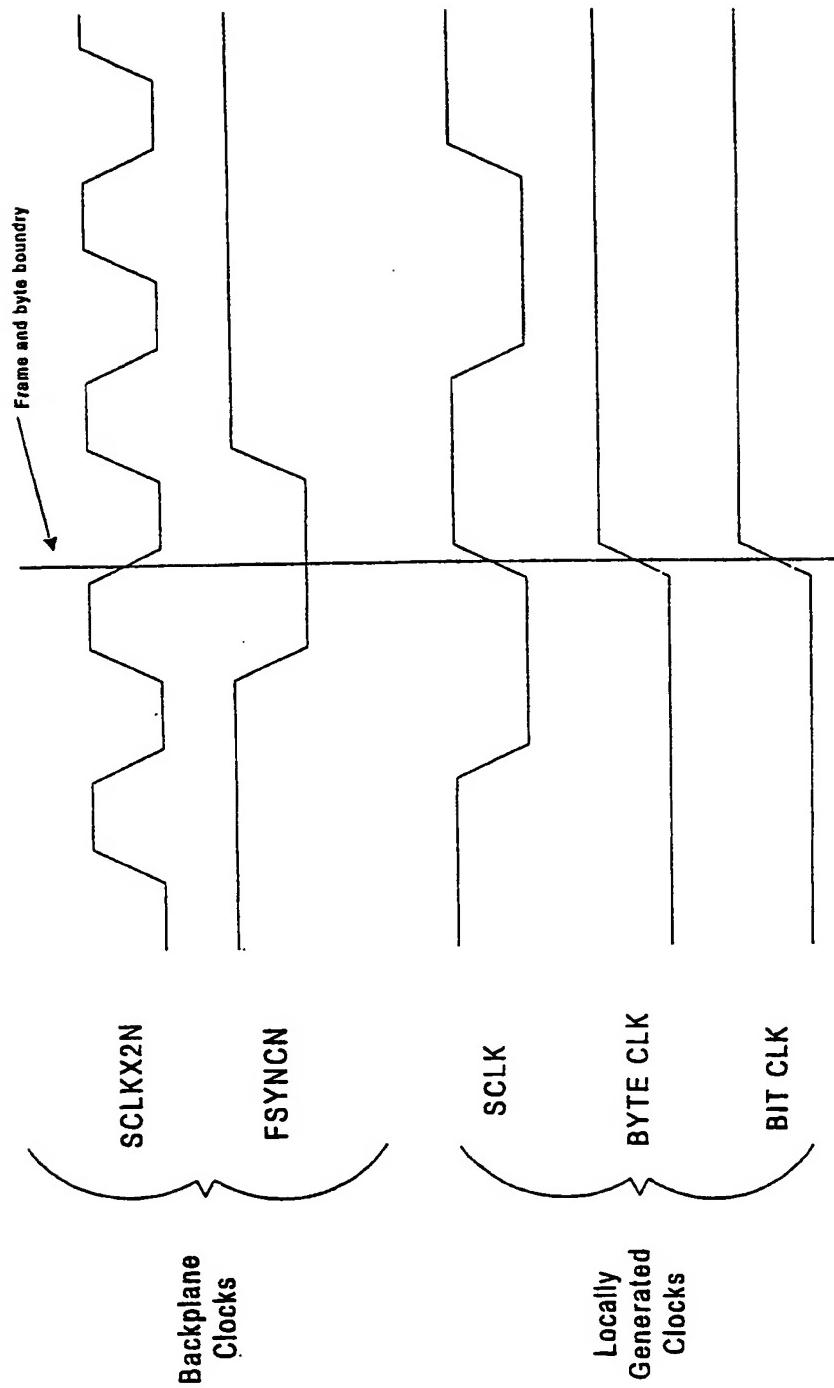


Fig. 5-1

42|58

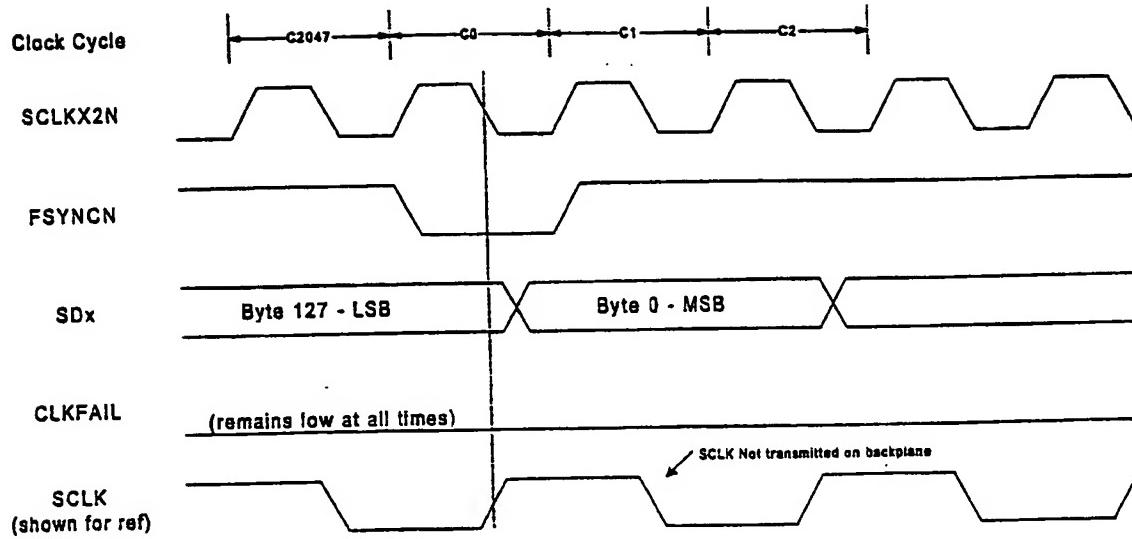


FIG. 52

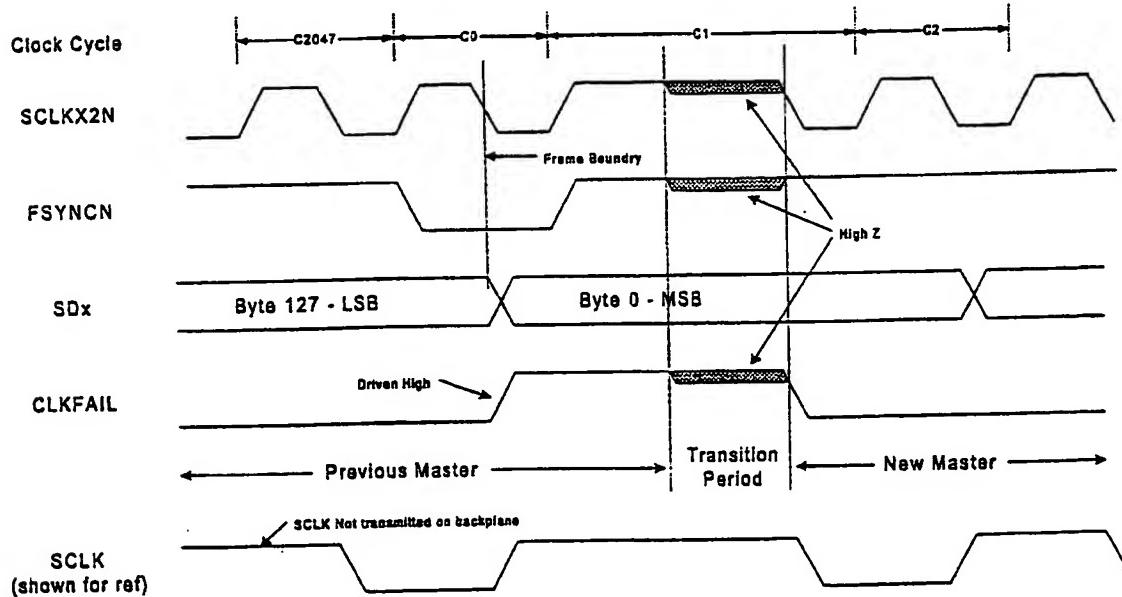


FIG. 53

43158

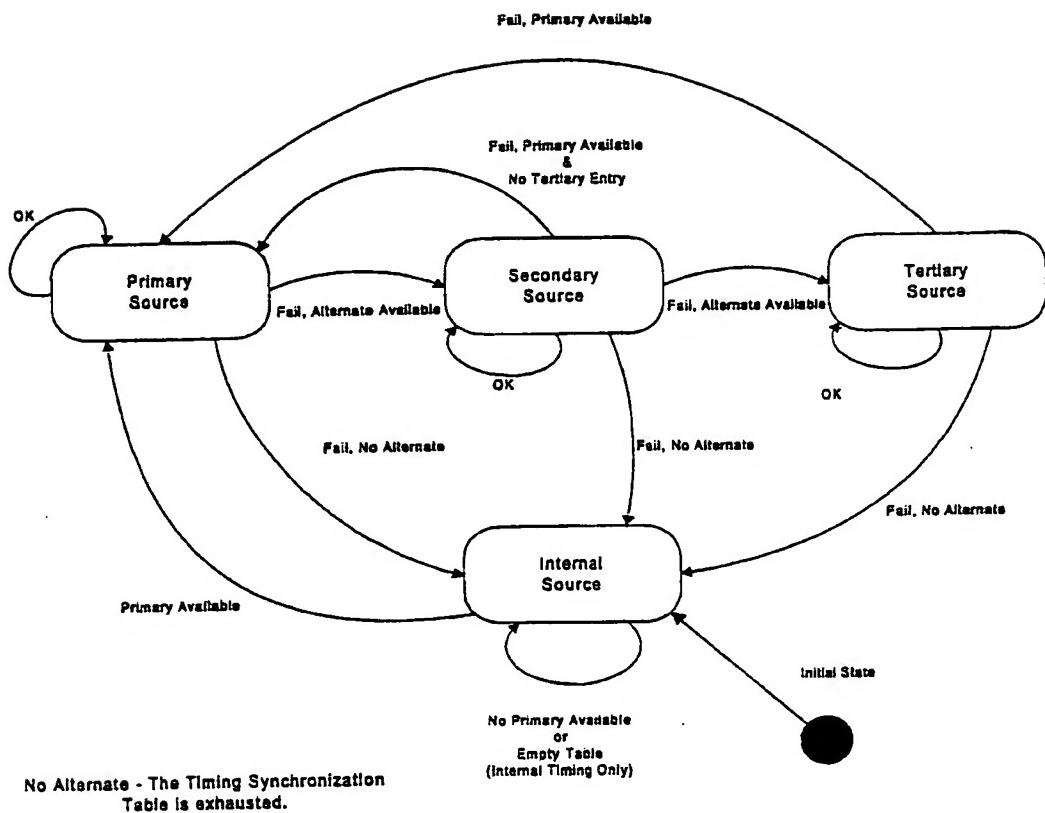


FIG. 59

44158

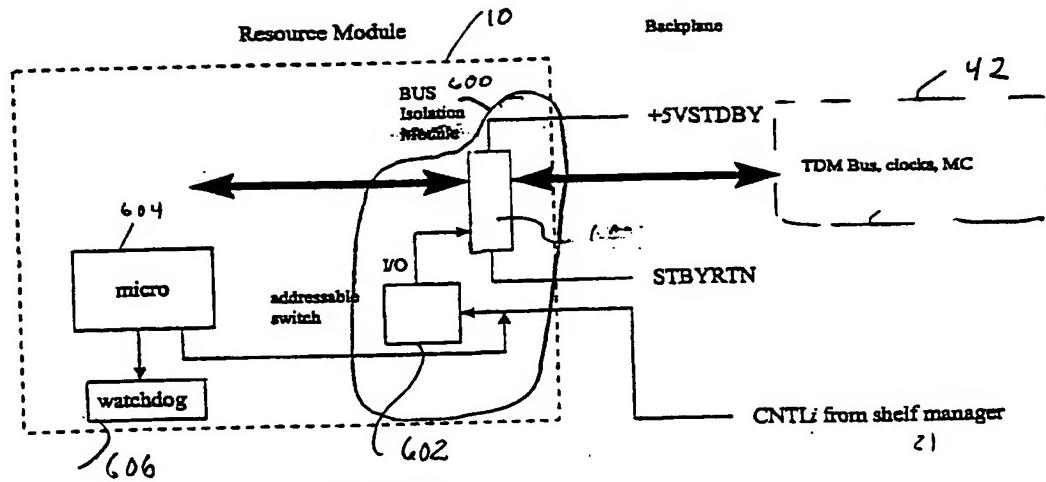


Figure 55

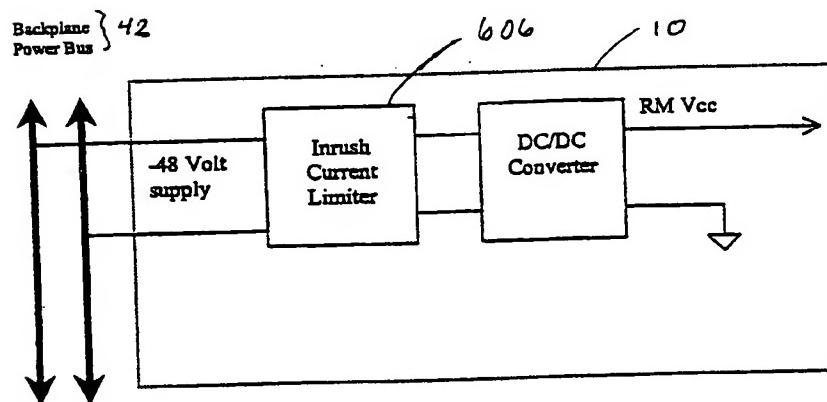


Figure 56

45 | 58

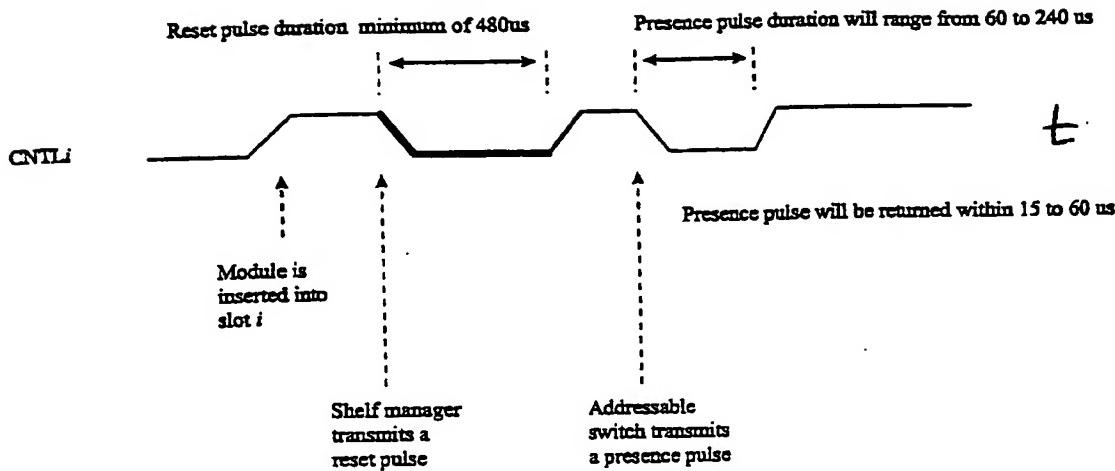


FIGURE 57

46/58

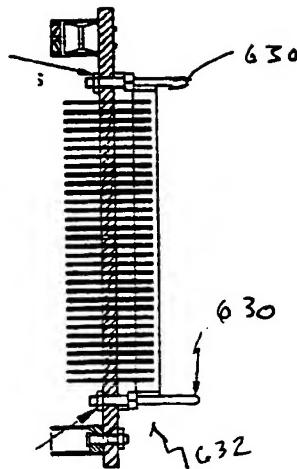


FIG 58

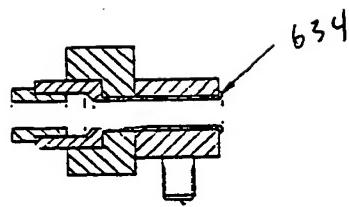


FIGURE 59

47/58

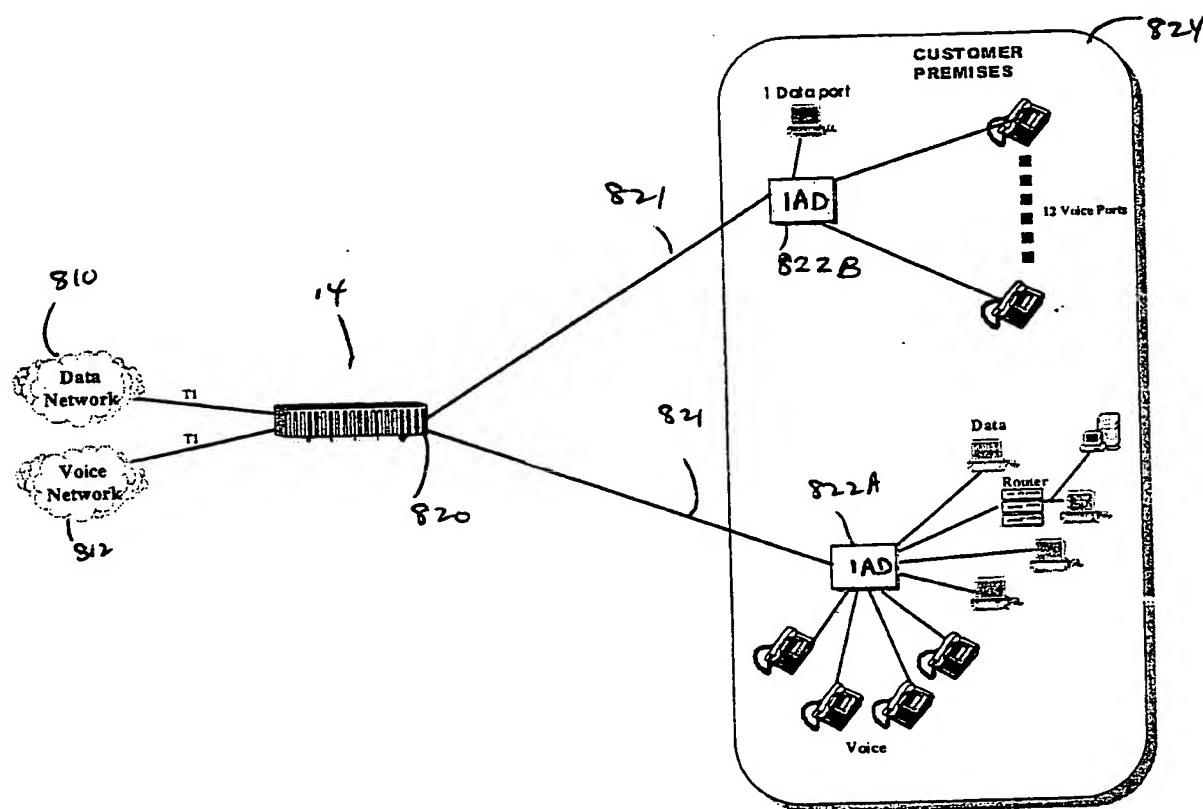


FIG. 60

48/58

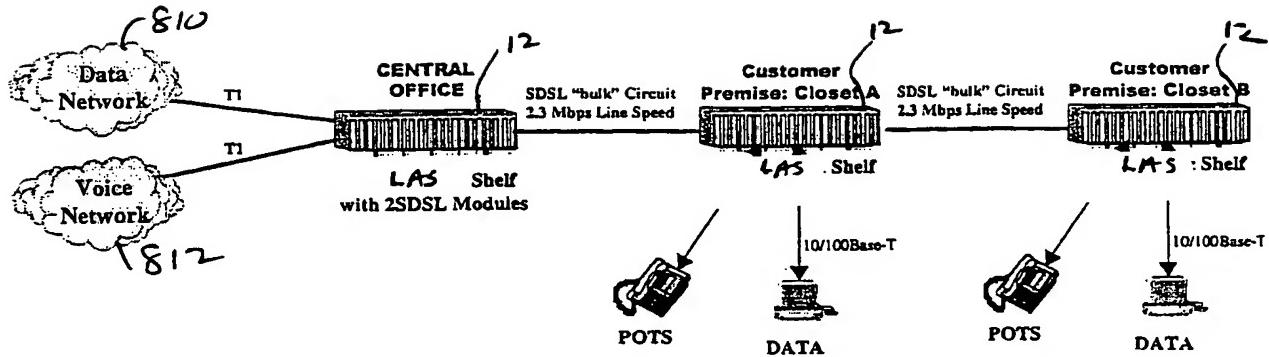
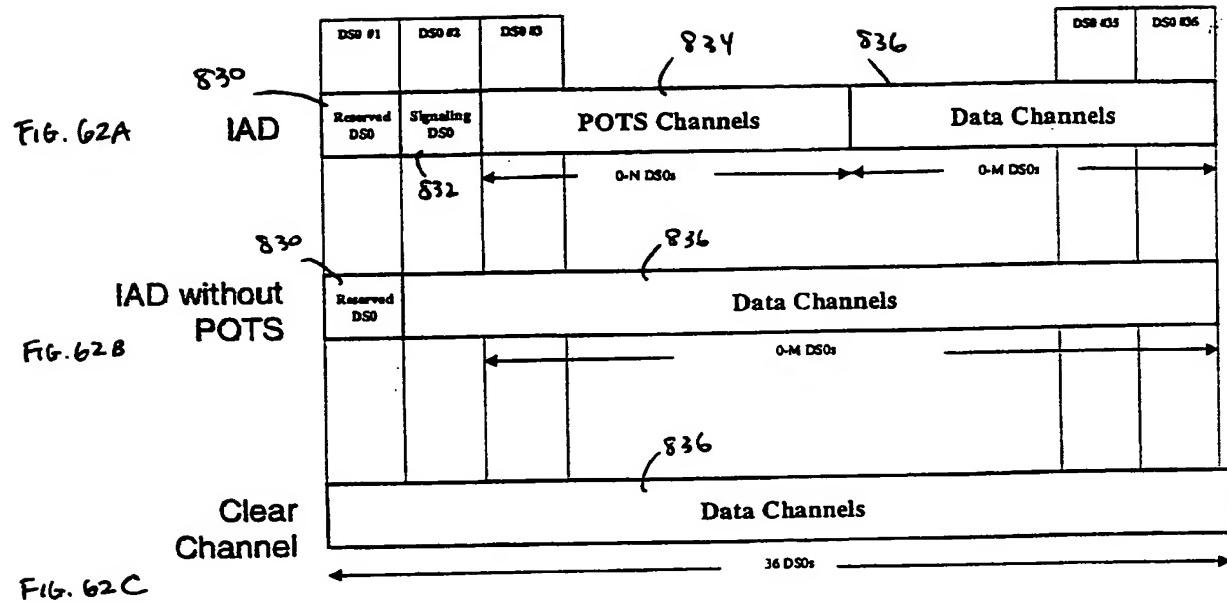


FIG. 61

49/58



50/58

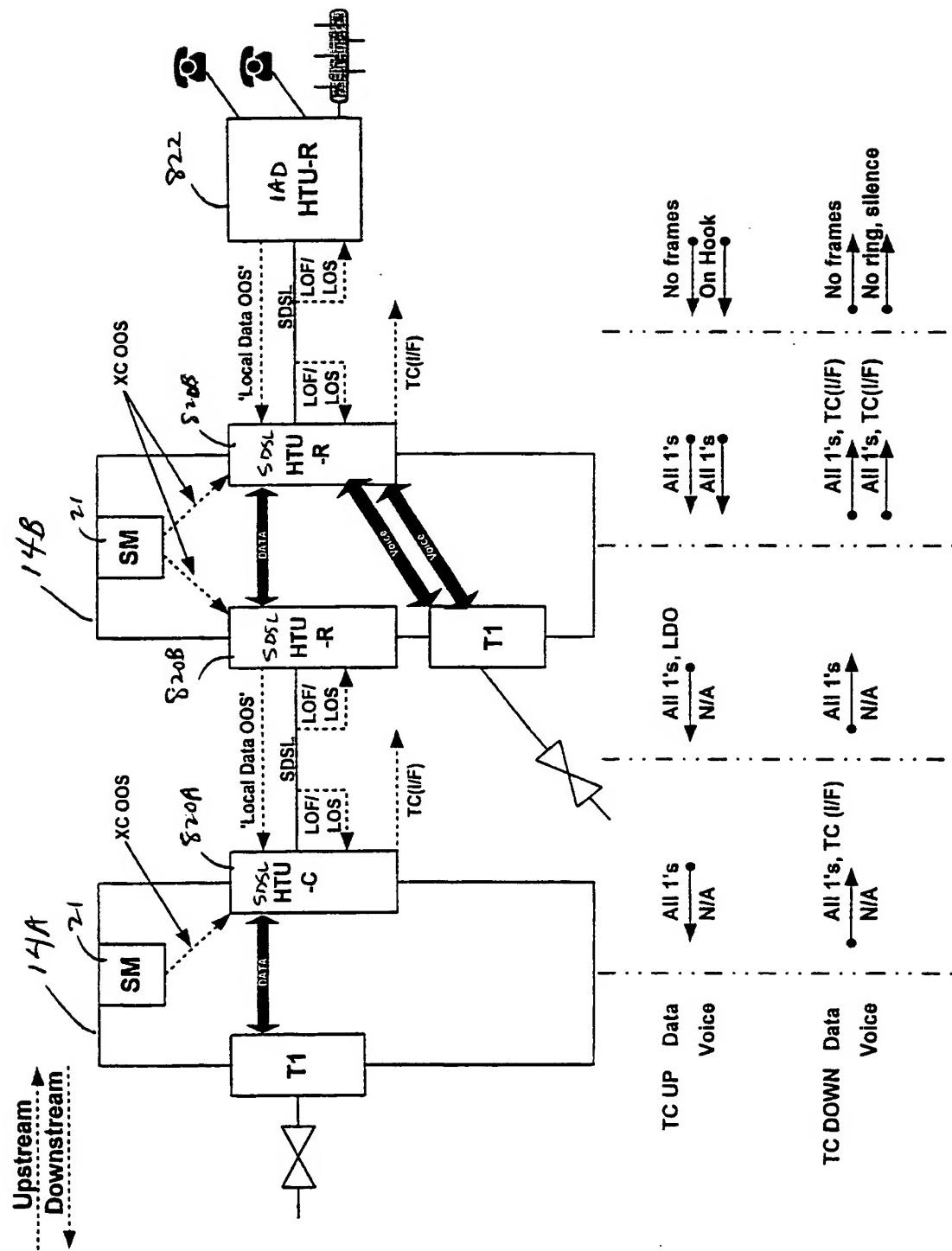


Fig. 63

51158

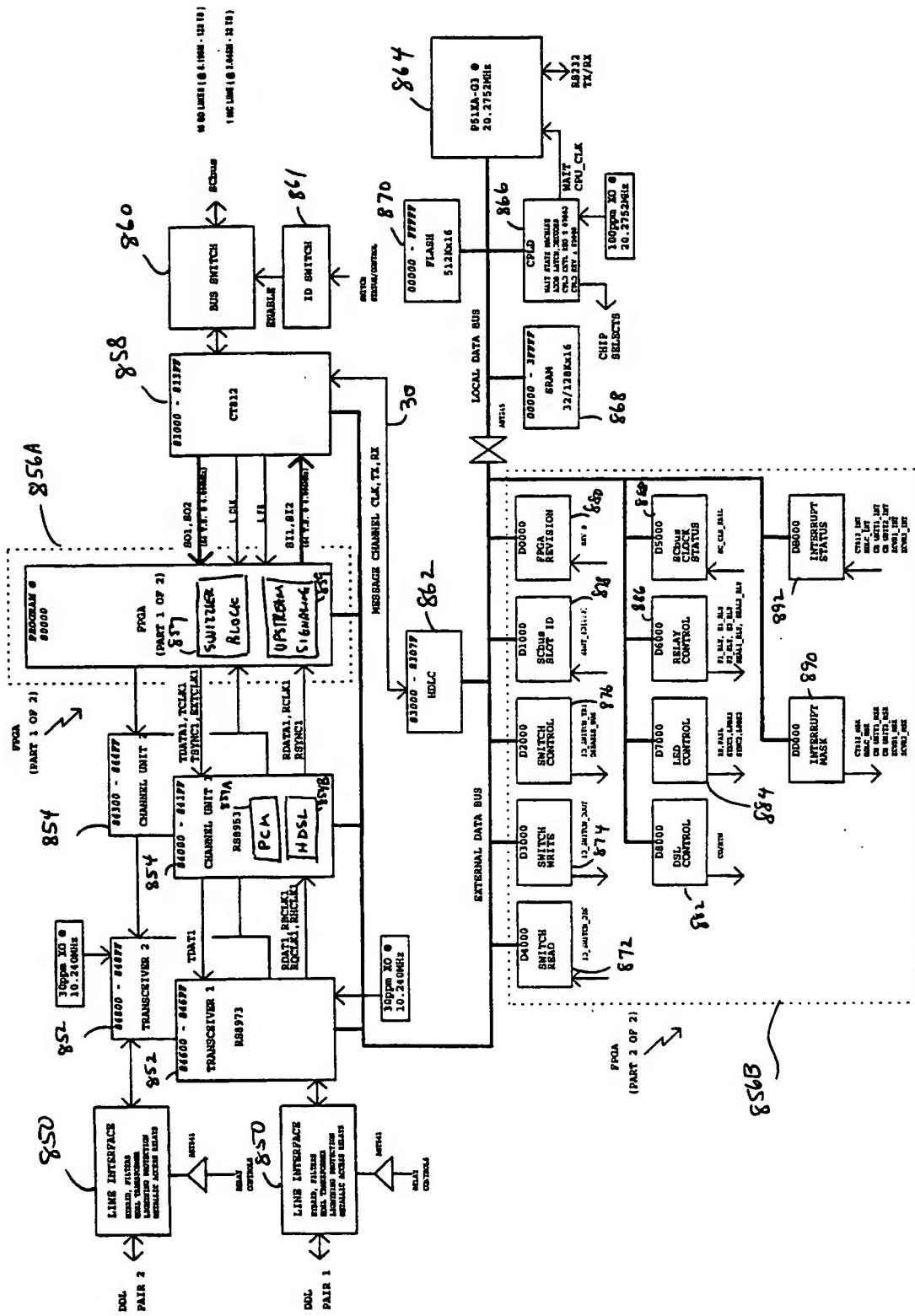


Fig. 64

5458

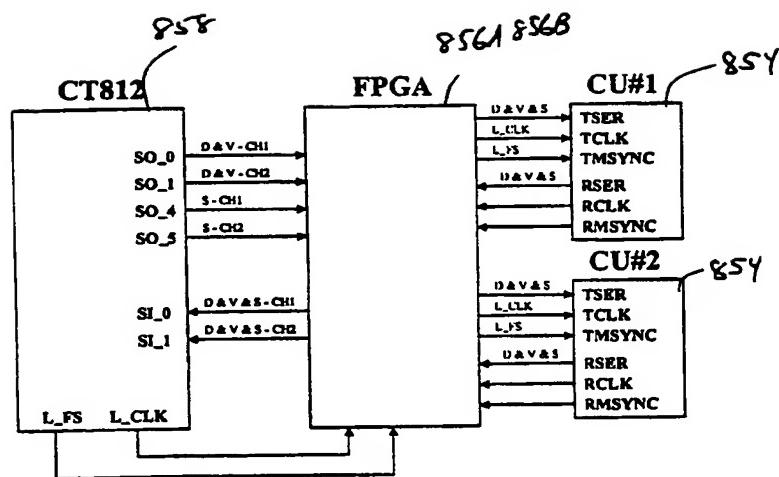


Fig. 65

53 / 58

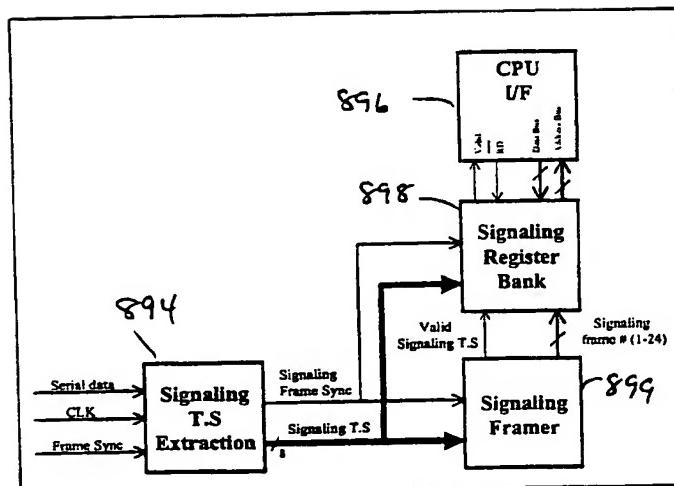


Fig. 66

54/58

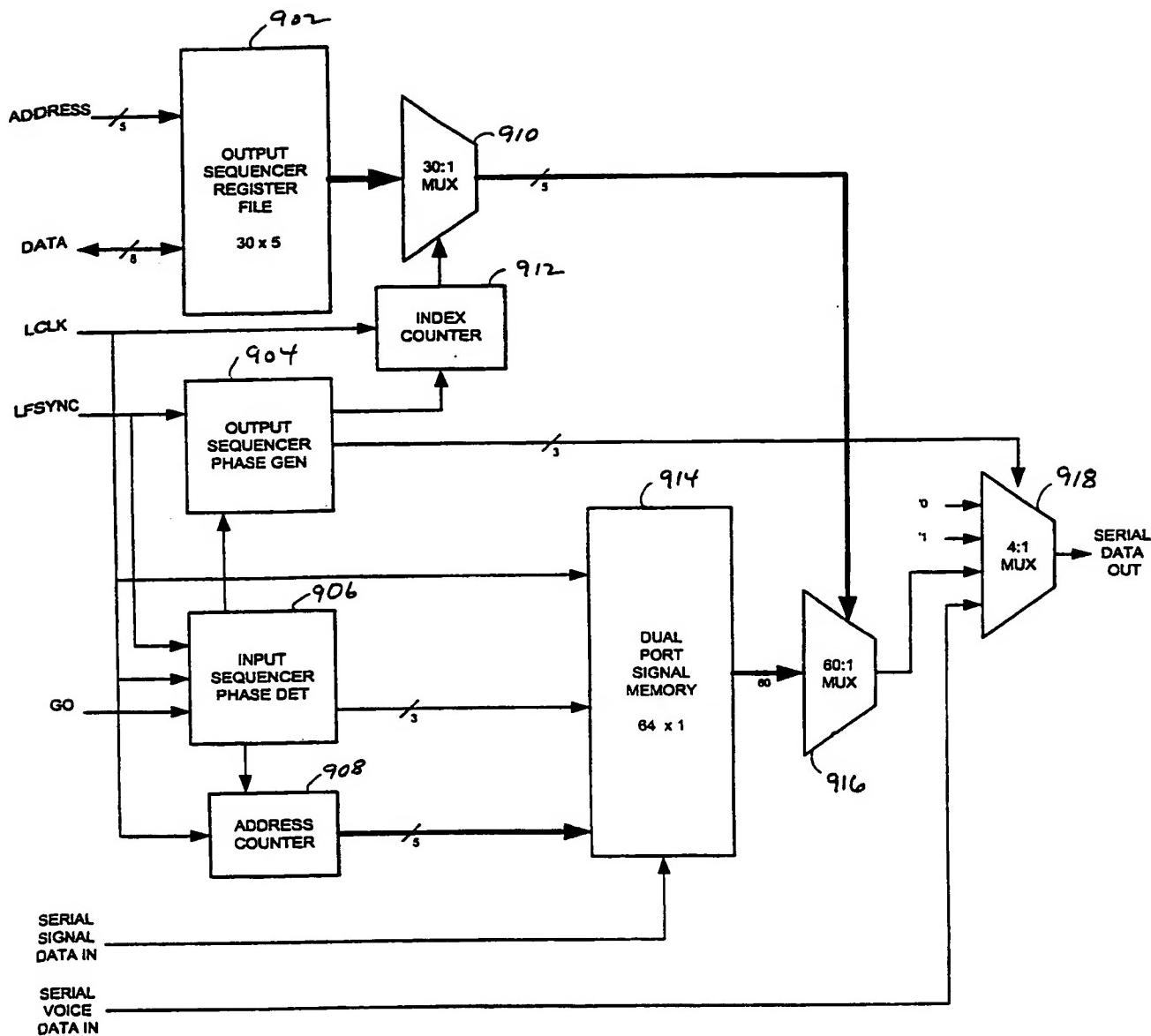


Fig. 67

55/58

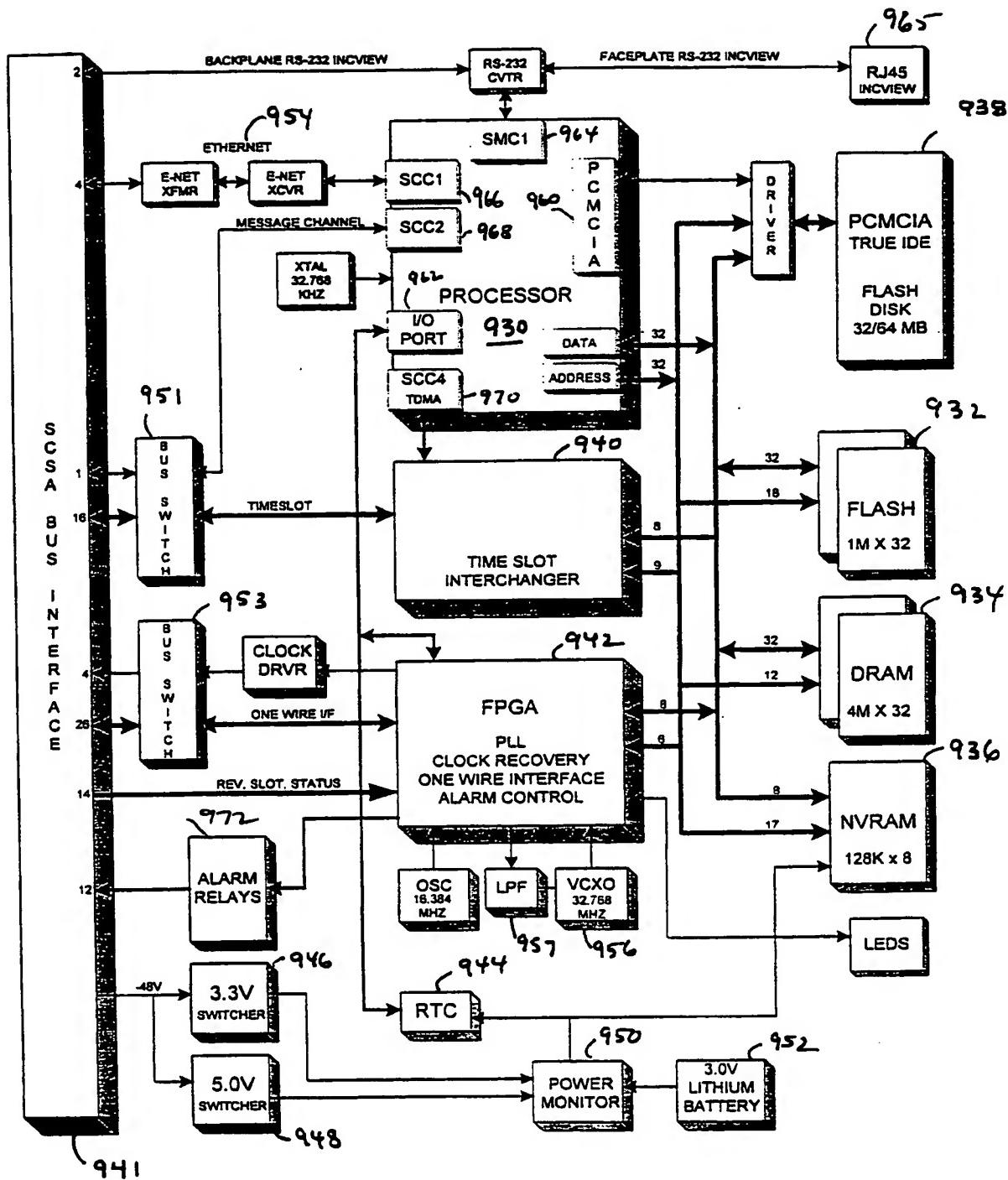


FIG. 68

56 / 58

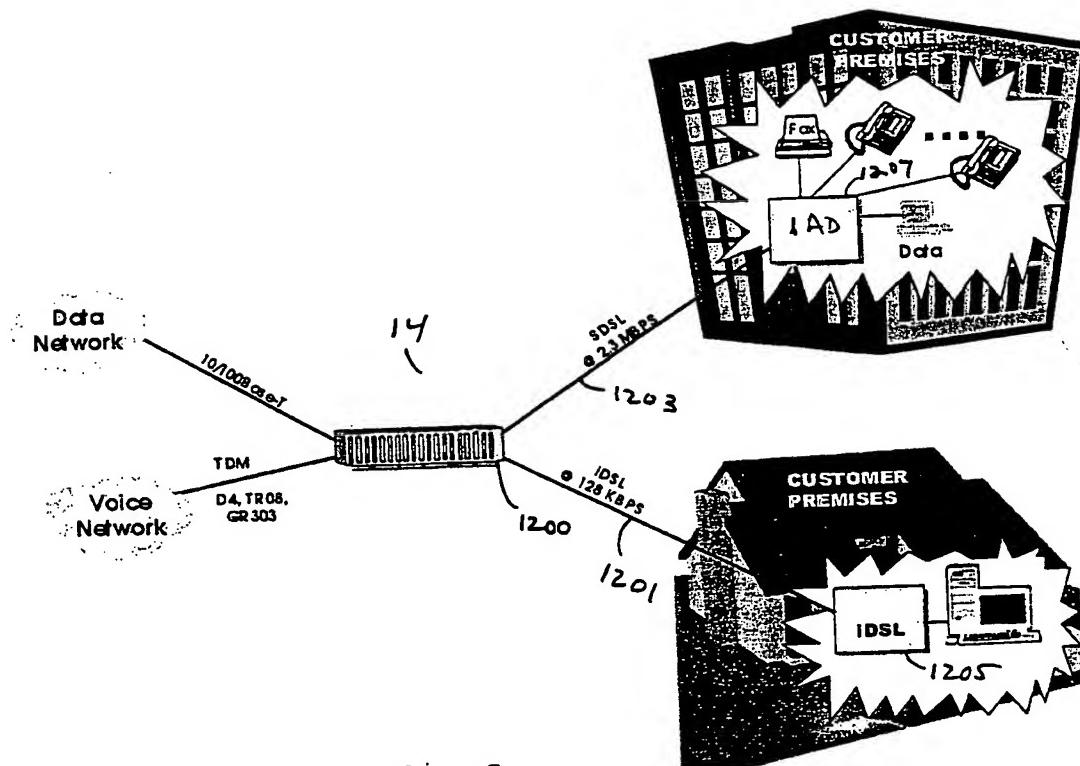


Fig. 69

57/58

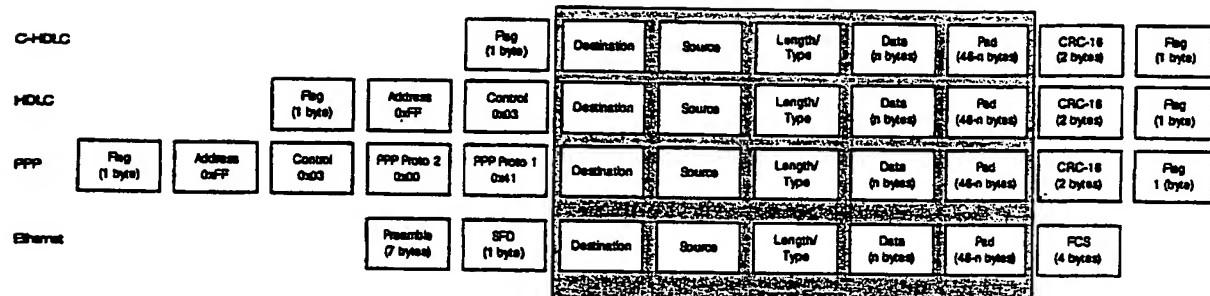


Fig. 70

58/58

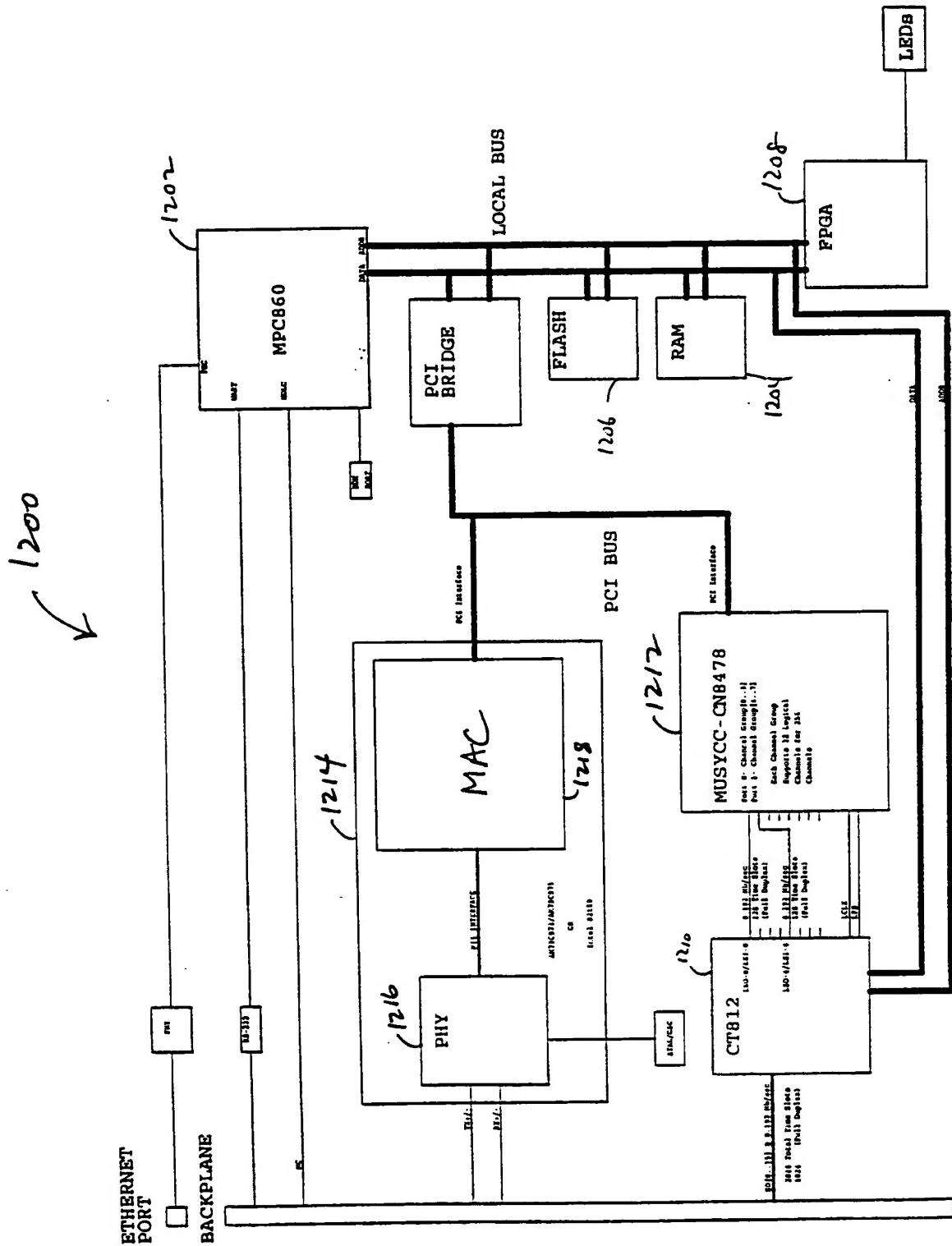


Fig. 11